Theoretical Understanding of Neural Network Optimization Landscape and Self-Supervised Representation Learning

by

Chenwei Wu

Department of Computer Science
Duke University

Date: _____

Approved:

_____
Rong Ge, Supervisor

_____
Kamesh Munagala

_____
Debmalya Panigrahi

_____
Jianfeng Lu

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Computer Science
in the Graduate School of Duke University
2023

Abstract

Theoretical Understanding of Neural Network Optimization Landscape and Self-Supervised Representation Learning

by

Chenwei Wu

Department of Computer Science
Duke University

Date: _____

Approved:

_____
Rong Ge, Supervisor

_____
Kamesh Munagala

_____
Debmalya Panigrahi

_____
Jianfeng Lu

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science in the Graduate School of Duke University
2023

Abstract

Neural networks have achieved remarkable empirical success in various areas. One key factor of their success is their ability to automatically learn useful representations from data. Self-supervised representation learning, which learns the representations during pre-training and applies learned representations in downstream tasks, has become the dominant approach for representation learning in recent years. However, theoretical understanding of self-supervised representation learning is scarce. Two main bottlenecks in understanding self-supervised representation learning are the big differences between pre-training and downstream tasks and the difficulties in neural network optimization. In this thesis, we present an initial exploration into analyzing the benefit of pre-training in self-supervised representation learning and two heuristics in neural network optimization.

The first part of this thesis presents our attempts to understand why the representations produced by pre-trained models are useful in downstream tasks. We assume we can optimize the training objective well in this part. For the over-realized sparse coding model with noise, we show that the masking objective used in pre-training ensures the recovery of ground-truth model parameters. For a more complicated log-linear word model, we characterize what downstream tasks can benefit from the learned representations in pre-training. Our experiments validate these theoretical results.

The second part of this thesis provides explanations about two important phenomena in the neural network optimization landscape. We first propose and rigorously prove a novel conjecture that explains the low-rank structure of the layer-wise neural network Hessian. Our conjecture is verified experimentally and can be used to tighten generalization bounds for neural networks. We also study the training stability and generalization problem in the learning-to-learn framework where machine learning algorithms are used to learn parameters for training neural networks. We rigorously proved our conjectures in simple models and empirically verified our theoretical results in our experiments with practical neural networks and real data.

Our results provide theoretical understanding of the benefits of pre-training for downstream tasks and two important heuristics of neural network optimization landscape. We hope these insights could further improve the performance of self-supervised representation learning approaches and inspire the design of new algorithms.

Contents

Abstract **iv**

List of Tables **xiii**

List of Figures **xiv**

Acknowledgements **xvii**

1 Introduction **1**

1.1 Theoretical Analysis of Self-Supervised Representation Learning . . . 4

1.2 Neural Network Optimization Landscape 5

1.2.1 Understanding Top Eigenspace of Neural Network Hessian . . 7

1.2.2 Learning-to-Learn for Tuning the Step Size 8

1.3 Other Related Works 9

1.4 Bibliographic Notes 11

1.5 Future Directions 11

2 Masking Helps Sparse Coding **13**

2.1 Introduction 13

2.1.1 Main Contributions and Outline 15

2.1.2 Related Work 16

2.2 Preliminaries and Setup 18

2.2.1 Background on Sparse Coding 18

2.2.2 Sparse Coding via Orthogonal Matching Pursuit 20
2.2.3 Conditions on the Data-Generating Process 21
2.3 Main Results 22
2.4 Experiments 26
2.4.1 Scaling Over-Realization 29
2.4.2 Scaling All Parameters 32
2.4.3 Analyzing Different Noise Levels 33
2.5 Conclusion 34
3 What Downstream Tasks Provably Benefit from Pre-Training? 36
3.1 Introduction 37
3.1.1 Our Contributions 38
3.1.2 Related Works 41
3.2 Problem Setup 43
3.3 Cases When Representations are Insufficient in Downstream Tasks . . 46
3.3.1 Downstream Tasks Sensitive to Words with Super-Small Probabilities 47
3.3.2 Representations are not Shift-Invariant 48
3.4 "Anchor Vector" Hypothesis and Empirical Verifications 49
3.4.1 "Anchor Vector" Hypothesis 49
3.4.2 Empirical Verification of "Anchor Vector" Hypothesis 51
3.4.3 Existence of Anchor Vector is not Trivial 52
3.5 Anchor Vector Guarantees Performance Transfer from Pre-Training to Downstream Tasks 54
3.5.1 Modifications to the Model 54
3.5.2 Assumptions 55

3.5.3 Main Theorem and Interpretations 56
3.5.4 Proof and Discussions for Main Theorem 58
3.5.5 Proof of Lemma 3.5.9 61
3.6 Conclusions and Future Work 66
4 Understanding Structure of Hessian for Neural Networks 68
4.1 Introduction 69
4.1.1 Outline 70
4.1.2 Related Works 72
4.2 Preliminaries and Notations 73
4.3 Decoupling Conjecture and Implications on the Structures of Hessian 76
4.3.1 Structure of Input Auto-Correlation Matrix $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ and Output Hessian $\mathbb{E}[\boldsymbol{M}]$ 77
4.3.2 Implications on the Eigenspectrum and Eigenvectors of Layer-Wise Hessian 77
4.4 Hessian Structure for Infinite Width Two-Layer ReLU Neural Network 78
4.5 Empirical Observation and Verification 82
4.5.1 Kronecker Approximation of Layer-Wise Hessian and Full Hessian 82
4.5.2 Low Rank Structure of $\mathbb{E}[\boldsymbol{M}]$ and $\boldsymbol{H}$ 83
4.5.3 Eigenspace Overlap of Different Models 84
4.6 Tighter PAC-Bayes Bound with Hessian Information 85
4.7 Limitations and Conclusions 87
5 Learning-to-Learn for Tuning the Step Size 88
5.1 Introduction 89
5.1.1 Our Results 90

5.1.2 Related Work 91
5.2 Preliminaries 93
5.2.1 Notations 93
5.2.2 Learning-to-Learn Framework 94
5.3 Alleviating Gradient Explosion/Vanishing Problems 95
5.3.1 Proof Sketch of Theorem 5.3.2 98
5.4 Generalization for Trained Optimizer 99
5.4.1 Proof Sketch for Train-by-Train 103
5.4.2 Proof Sketch for Train-by-Validation 106
5.5 Experiments 108
5.6 Conclusions 111
6 Conclusion **114**
A Supplementary Materials for Chapter 2 **115**
A.1 Full Proofs 115
A.1.1 Proof of Theorem 2.3.2 115
A.1.2 Proof of Theorem 2.3.6 119
A.1.3 Auxiliary Lemmas 125
B Supplementary Materials for Chapter 3 **126**
B.1 Proof of Theorem 3.3.1 126
B.2 Detailed Proofs of Auxilliary Lemmas 128
B.3 Experiment Details 133
C Supplementary Materials for Chapter 4 **137**
C.1 Detailed Derivations 137

C.1.1 Derivation of Hessian 137
C.1.2 Approximating Weight Hessian of Convolutional Layers 140
C.2 Main Proof 143
C.2.1 Preliminaries 143
C.2.2 Detailed Proof 146
C.3 Structure of Dominating Eigenvectors of the Full Hessian. 201
C.4 Computation of Hessian Eigenvalues and Eigenvectors 206
C.5 Detailed Experiment Setup 207
C.5.1 Datasets 207
C.5.2 Network Structures 208
C.5.3 Training Process and Hyperparameter Configuration 211
C.6 Additional Empirical Results 213
C.6.1 Low Rank Structure of Auto-Correlation Matrix $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ . . . 213
C.6.2 Eigenspace Overlap Between Different Models 217
C.6.3 Eigenvector Correspondence 219
C.6.4 Structure of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ and $\mathbb{E}[\boldsymbol{M}]$ During Training 221
C.7 Additional Explanations 223
C.7.1 Dominating Eigenvectors of Layer-Wise Hessian are Low Rank 223
C.7.2 Eigenspace Overlap of Different Models 223
C.7.3 Batch Normalization and Zero-Mean Input 229
C.7.4 Outliers in Hessian Eigenspectrum 231
C.8 Computing PAC-Bayes Bounds with Hessian Approximation 234
D Supplementary Materials for Chapter 5 **252**

D.1 Proofs for Section 5.3 – Alleviating Gradient Explosion/Vanishing Problem for Quadratic Objective . . . 253
D.1.1 Meta-Gradient Vanishing/Explosion . . . 253
D.1.2 Alleviating Meta-Gradient Vanishing/Explosion . . . 255
D.2 Proofs of Train-by-Train v.s. Train-by-Validation (GD) . . . 261
D.2.1 Overall Proof Strategy . . . 264
D.2.2 Train-by-Train (GD) . . . 266
D.2.3 Train-by-Validation (GD) . . . 285
D.3 Proofs of Train-by-Train with Large Number of Samples (GD) . . . 318
D.3.1 Upper Bounding $\widehat{F}_{TbT}(2/3)$ . . . 322
D.3.2 Lower Bounding $\widehat{F}_{TbT}$ for $\eta \in (\widehat{\eta}, \infty)$ . . . 326
D.3.3 Generalization for $\eta \in [0, \widehat{\eta}]$ . . . 329
D.3.4 Proofs of Technical Lemmas . . . 332
D.4 Proofs of Train-by-Train v.s. Train-by-Validation (SGD) . . . 335
D.4.1 Train-by-Train (SGD) . . . 338
D.4.2 Train-by-Validation (SGD) . . . 344
D.5 Tools . . . 362
D.5.1 Norm of Random Vectors . . . 362
D.5.2 Singular Values of Gaussian Matrices . . . 362
D.5.3 Johnson-Lindenstrauss Lemma . . . 363
D.6 Experiment Details . . . 364
D.6.1 Experiment Settings . . . 364
D.6.2 Additional Results . . . 368

Bibliography **372**

Biography **390**

List of Tables

3.1 Comparison between MSE of log bulk partition function (except top-100) and log partition function 53

4.1 Optimized PAC-Bayes bounds using different methods. 86

B.1 Parameters and Performances of Languge Models 134

B.2 Mean squared approximation error of log bulk partition function for different models and different k. 136

C.1 Datasets 207

C.2 Structure of F-200^2 on MNIST 208

C.3 Structure of LeNet5 on CIFAR-10 209

C.4 Structure of LeNet5-BN on CIFAR-10 210

C.5 Squared dot product and spectral ratio for fully connected layers in a selection of network structures and datasets. 215

C.6 Squared dot product and spectral ratio for convolutional layers in the selection of network structures and datasets in Table C.5. 216

C.7 Structure of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ for BN networks 229

C.8 Full PAC-Bayes bound optimization results 240

D.1 Whether the implementation converges for different t (fixed $\eta_0 = 0.1$) 369

D.2 Whether the implementation converges for different η_0 (fixed $t = 40$) . 369

List of Figures

2.1 Comparison of Algorithm 0 (Baseline) and Algorithm 0 (Masking) under the settings of Sections 2.4.1. 31

2.2 Comparison of Algorithm 0 (Baseline) and Algorithm 0 (Masking) under the settings of Section 2.4.2 (all parameter scaling). 33

2.3 Comparison of Algorithm 0 (Baseline) and Algorithm 0 (Masking) under the settings of Section 2.4.3 (noise scaling). 33

3.1 Mean squared error of log bulk partition function with different k in different models. 52

3.2 Singular values of dictionary matrices. 53

3.3 Histogram of log original and bulk partition functions 53

4.1 Some interesting observations on the structure of layer-wise Hessians. 71

4.2 Comparison between the approximated and true layer-wise Hessian of F-200[2]. 83

4.3 Eigenspectrum of the layer-wise output Hessian and the layer-wise weight Hessian. 83

4.4 Overlap between the top k dominating eigenspace of different independently trained models. 84

5.1 Meta training trajectory for η ($t = 80$, $\eta_0 = 0.1$). 108

5.2 Training and testing RMSE for different σ values (500 samples) . . . 109

5.3 Training and testing RMSE for different samples sizes ($\sigma = 1$) 110

5.4 The test accuracy of different optimizers in various settings. 112

5.5 Training accuracy for 1000 samples and 20% noise (same setting as in Figure 5.4(b)) . . . 113

B.1 ℓ_2-norms of atoms . . . 135

B.2 Cosine Similarity between two random atoms . . . 135

B.3 Histogram of log partition function . . . 135

B.4 Histogram of log bulk partition function . . . 136

C.1 Top 50 Eigenvalues and Eigenspace approximation for full Hessian . . 205

C.2 Eigenspace overlap of different models of LeNet5 trained with different hyperparameters. . . . 217

C.27 Optimized posterior variance, $\boldsymbol{s}$. (fc1:T-200^2, trained on MNIST), the horizontal axis is ordered with decreasing eigenvalues. . . . 239

C.3 Top Eigenspace overlap for varients of VGG11 on CIFAR10 and CIFAR100 . . . 241

C.4 Top Eigenspace overlap for variants of ResNet18 on CIFAR100 242

C.5 Top eigenspace overlap for layers with an early low peak. 243

C.6 Top eigenspace overlap for layers with a delayed peak. 243

C.7 Heatmap of Eigenvector Correspondence Matrices for fc1:LeNet5. . . 244

C.8 Eigenvector Correspondence for fc1:LeNet5. (m=120) 244

C.9 Eigenvector Correspondence for fc2:LeNet5. (m=84) 245

C.10 Eigenvector Correspondence for conv1:LeNet5. (m=6) 245

C.11 Eigenvector Correspondence for conv2:LeNet5. (m=16) 245

C.12 Eigenvector Correspondence with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ for conv1:VGG11. (m=64) 245

C.13 Eigenvector Correspondence with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ for conv2:VGG11. (m=128) 246

C.14 Eigenvector Correspondence with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ for conv3:VGG11. (m=256) 246

C.15 Top eigenvalues of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ along training trajectory. (fc1:LeNet5) . . 246

C.16 Top eigenvalues of $\mathbb{E}[\boldsymbol{M}]$ along training trajectory. (fc1:LeNet5) . . . 247

C.17 Ratio between top singular value and Frobenius norm of matricized dominating eigenvectors. 247

C.18 Top eigenspace overlap for the final fully connected layer. 247

C.19 Eigenspace overlap, eigenspectrum, and cropped (upper 20×20 block) eigenvector correspondence matrices for fc2:F-200^2 (MNIST) 248

C.20 Eigenspace overlap, eigenspectrum, and cropped (upper 50×50 block) eigenvector correspondence matrices for conv2:VGG11-W200 (CIFAR10) 248

C.21 Eigenspace overlap, eigenspectrum, and cropped (upper 50×50 block) eigenvector correspondence matrices for conv2:VGG11-W200 (CIFAR10) 249

C.22 Eigenspectrum and Eigenvector correspondence matrices with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ for LeNet5-BN. 249

C.23 Eigenspace overlap of different models of LeNet5-BN. 250

C.24 Comparison between the true and approximated layer-wise Hessians for LeNet5-BN. 250

C.25 Logit clustering behavior of $\boldsymbol{\Delta}$ and $\boldsymbol{\Gamma}$ at initialization (fc1:T-200^2) . . 251

C.26 Class clustering behavior of $\boldsymbol{\Delta}$ and $\boldsymbol{\Gamma}$ at minimum. (fc1:T-200^2) . . . 251

D.1 Training and testing accuracy for different models (all samples, 20% noise) 370

D.2 Training and testing accuracy for different models (12000 samples, no noise) 371

D.3 Training and testing accuracy for different models (12000 samples, 20% noise) 371

Acknowledgements

I would like to extend my deepest gratitude to my Ph.D. advisor Rong Ge. Rong is always a source of invaluable guidance, support, and encouragement during my Ph.D. study. When I felt depressed doing research remotely during the second and third years, it was his care and support that took me out of the depression. I also learned from him the confidence, perseverance, and optimism when facing challenges. I was given much freedom in my research, and he has always been supporting me in both high-level directions and technical details. He also provides advice on my career choice, which influences me a lot.

I thank Erran Li for hosting me as an intern at Amazon. I learned a lot from his viewpoints and gained much industrial experience from this internship. I also thank my mentors and collaborators Stefano Ermon, Patrick Haffner, and Zaiwei Zhang from Amazon.

I was fortunate to be hosted by Yang Yuan as a research intern at Haihua Institute. This experience expanded my horizons.

I would also like to thank the other members of my dissertation committee: Kamesh Munagala, Debmalya Panigrahi, and Jianfeng Lu for their valuable advice on my research.

I am grateful to Andrew Chi-Chih Yao for founding Yao Class where I was an undergraduate student. The courses and projects there led me to the research path.

I also thank Jason Lee and Liwei Wang for advising my undergraduate research.

I would also like to thank the rest of my collaborators, without whom this thesis would not be possible. An incomplete list would be: Yu Cheng, Muthu Chidambaram, Chenzhuang Du, Rong Ge, Yuzheng Hu, Holden Lee, Jason Lee, Tengyu Ma, Annie Wang, Xiang Wang, Yikai Wu, Shuai Yuan, Yang Yuan, and Xingyu Zhu.

Thanks to the helpful staff at Duke Computer Science, especially Marilyn Butler. I also thank the friends I have made at Duke: Hanrui, Jingrong, Jiyao, Kangning, Mo, Muthu, Shihan, Xiang, Xiao, and Zhengjie, for their companionship throughout this journey.

I thank Chenzhuang, Haowei, Jianhao, Jiaye, Kaifeng, Tianle, Zhou, and Zixin. Their enthusiasm helped me overcome the hard times.

Finally, I would like to thank my parents, Shaozhong Wu and Lei Zhu, for their unconditional support and love. They are always giving me the freedom to make my own decisions. I would also like to thank my girlfriend Yan Meng for bringing passion to my life.

1

Introduction

Neural networks have demonstrated astonishing success across a wide range of domains, including computer vision (He et al., 2016a), natural language processing (Vaswani et al., 2017), speech recognition (Gulati et al., 2020), reinforcement learning (Mnih et al., 2015), and many more (Bommasani et al., 2021). In practice, neural networks can automatically learn useful representations from data and use these representations for various specific tasks. These representations can usually achieve better performances in the tasks than handcrafted ones. However, the theoretical understanding of these learned representations is very limited. This has led to a dependence on trial-and-error approaches which are time-consuming and resource-intensive. A better theoretical analysis of these representations will help us better understand neural networks, and as a result, help us improve their empirical performances and even unlock new applications.

The representations of neural networks can be learned in supervised, unsupervised, or self-supervised ways. In supervised representation learning, each datum is given a label, i.e., the correct output, and the goal of the representations is to

help the model predict the labels more accurately. Unsupervised learning, on the other hand, does not require labels and learns the representations by finding the underlying patterns or structures of the data. Self-supervised learning is a variant of unsupervised learning. It deals with unlabeled data but transforms the problem into a supervised task by creating "labels" from the data themselves. For example, contrastive learning (Oord et al., 2018) asks the model to pull representations for similar data closer and push representations for different data farther. As another example, in masking (Devlin et al., 2018), the model is trained to predict a portion of its input data given the other portion. This representation-learning process using unlabeled data is called pre-training. In most cases, the representations learned in pre-training will be used as input in a supervised task with few labeled data. The supervised task is called the downstream task.

Self-supervised learning is especially useful when there are abundant unlabelled data but few labeled data for the task. For instance, people have unlabeled text corpora that contain billions of sentences (Raffel et al., 2020a), but may only have a thousand labeled sentences for a sentiment classification task. It is empirically shown that the representations learned by self-supervised learning achieve state-of-the-art performances in these cases (Chen et al., 2020) — more importantly, these representations are much more transferable than before. Representations learned in the pre-training stage are usually useful for multiple tasks (Devlin et al., 2018), even tasks in other modalities, e.g., pre-trained language model representations can help image generation (Ramesh et al., 2022).

Theoretically understanding self-supervised representation learning is very challenging for at least two reasons: Firstly, the pre-training task is usually quite different from the downstream tasks. The neural networks are usually trained to predict part

of the input data during pre-training, but the downstream task can be classification, detection, etc. This difference in tasks also usually results in a difference in their objective functions and sometimes a difference in data distributions. Secondly, the process of learning the parameters of neural networks is complicated. These parameters are usually learned by minimizing a loss function using local search algorithms. Since the number of parameters in a neural network can be at the scale of billions and even trillions (Fedus et al., 2021), the parameter space is extremely high-dimensional. Furthermore, neural networks often have deep layered structures with non-linearities in between, resulting in a highly non-convex and non-smooth loss landscape.

This thesis represents our initial attempts to theoretically understand the benefit of pre-training in self-supervised learning and several heuristics in neural network optimization. In the first part of this thesis, we try to understand why the representations learned in the pre-training stage benefit the downstream tasks. In this part, we assume the optimization is successful and we get a model with low pre-training loss. We then focus on how to choose the proper pre-training loss that allow pre-trained representations to be useful. As a concrete example, we will prove that masking, which is a commonly used objective in self-supervised learning, ensures the recovery of ground-truth parameters in an over-realized sparse coding model. Generalizing the insights to more complicated models, we will also provide some necessary and sufficient conditions for the downstream task to provably benefit from pre-trained representations in large language models. More detail about these results will be provided in Section 1.1.

In the second part of the thesis, our goal is to understand the optimization of neural networks. Since the optimization landscape of neural networks is very complicated, current global convergence results for training neural networks are restricted

to simple models and shallow neural networks. However, neural networks used in practice are much more complicated. Therefore, people use many heuristics to accelerate training or make the learned model perform better on unseen data. In the second part of this thesis, we investigate two such heuristics and provide theoretical insights about them. We will discuss more details in Section 1.2.

1.1 Theoretical Analysis of Self-Supervised Representation Learning

Our first goal is to understand why the downstream tasks can benefit from the representations learned in the pre-training stage. To do that, we need realistic and reasonable connections between the pre-training and downstream tasks. A recent line of work (Lee et al., 2021; Tosh et al., 2021b,c; Wei et al., 2021) assumes that they share the same latent representations: the shared representations generate the pre-training data via a latent variable model (such as Hidden Markov Model), and at the same time decide the labels for the downstream tasks. This is also the framework we will use in our analysis. We extend the previous results to new models. Moreover, our studies give both the success and failure cases for self-supervised learning approaches, providing a more complete understanding of the benefit of pre-training.

In Chapter 2, we analyze why the representations learned using a self-supervised approach during pre-training can perform better than traditional methods. We study masking, where the models are asked to predict part of the data (masked part) from the other part (unmasked part). As an example, we focus on the sparse coding problem with an over-realized dictionary as the generative model for the pre-training data. Existing theoretical results are limited to noiseless data (Sulam et al., 2022). Under the noisy data setting, we proved that minimizing the standard dictionary learning objective can fail to recover the ground-truth parameters, but the masking

objective has the ground-truth parameters as its optimal solution. In other words, a proper choice of pre-training objective helps us learn useful representations. Our experiments validate our theoretical results and showed that the learned parameters from masking can achieve better recovery performance than the traditional method.

A major bottleneck in existing works (Arora et al., 2016; Wei et al., 2021), including the result in Chapter 2, is that they are limited to simple and well-understood probabilistic models. In Chapter 3, we give initial results in understanding the benefit of pre-training for much more complicated models. In this work, we open the black box of huge models at the last layer. Specifically, we only assume that the last layer of the generative model for pre-training data is a log-linear probabilistic model. The network before the last layer can be arbitrarily complicated. In this setting, we first find two necessary conditions for the downstream tasks to be guaranteed to benefit from the representations: the insensitivity to extremely rare words and the resistance to shift-invariance of the softmax function. We also proposed, empirically verify, and theoretically proved a sufficient condition for guaranteeing good downstream performance, which is the existence of an "anchor vector" in the representation space.

1.2 Neural Network Optimization Landscape

The loss function is a function of network parameters measuring the discrepancy between the network outputs and our desired outputs. Its landscape – locations and loss values for various local optimal solutions and saddle points – plays a very important role in determining the quality of the final learned parameters. The commonly used algorithms for neural networks are first-order local search algorithms, such as gradient descent and its variants. These algorithms iteratively take small steps in

locally downhill directions on the loss landscape and are guaranteed to learn the optimal parameters for convex functions. However, the optimization landscape of neural networks is highly non-convex and usually has many spurious local minima that are problematic for local search algorithms (Safran and Shamir, 2018). Despite the complicated landscape, these algorithms in practice often find a point with a low loss value. Moreover, the number of parameters in neural networks is usually larger than the number of training samples, resulting in many global minima on the loss landscape (Soudry and Hoffer, 2017). Although only a few global minima perform well on unseen data (Keskar et al., 2017), the algorithms often converge to these minima. The two phenomena are still far from being fully understood in theory.

The theoretical results capturing the global convergence of neural network optimization are usually limited to unrealistic models, e.g., linear neural networks (Kawaguchi, 2016) or networks with near-infinite widths (Allen-Zhu et al., 2019; Mei et al., 2018). In this thesis, we want to understand the optimization of practical deep neural networks, whose optimization landscapes are much more complicated than the ones in these theoretical results (Li et al., 2018). People usually use several heuristics in training a practical neural network to help with its optimization and its generalization to unseen data. Therefore, understanding these heuristics is crucial for understanding the optimization of practical deep neural networks. The theoretical insights gained from the understanding may help improve the optimization performance and the design of new algorithms. In this section, we improve the theoretical understanding of neural network optimization landscape by theoretically understanding two such important heuristics about the optimization landscape. We also empirically show that our new understanding can help improve the performances of practical neural networks.

1.2.1 Understanding Top Eigenspace of Neural Network Hessian

Neural network Hessian captures important properties of the optimization landscape. Hessian provides second-order information of the loss, e.g., the local optimality of a critical point and the curvature in a local area. These information can help us analyze the behavior of local search algorithms, including their convergence speeds, whether they can get stuck, etc. Besides, Hessian can be used to design better algorithms for neural network training, such as L-BFGS (Liu and Nocedal, 1989) and K-FAC (Martens and Grosse, 2015). Furthermore, extensive experiments have shown that the Hessian around a local minimum on the optimization landscape is closely related to the generalization performance of the corresponding parameters (Jiang et al., 2019). Despite such importance, little theory is known about the structure of neural network Hessian.

In Chapter 4, we are going to provide a theoretical analysis of the low-rank structure of the neural network Hessian, a phenomenon that has been observed in previous works (Sagun et al., 2018). We propose a decoupling conjecture that decomposes the layer-wise Hessian of a neural network as the Kronecker product of two smaller matrices, and rigorously prove the conjecture for randomly initialized 2-layer networks. This decoupling not only explains the previous observations about Hessian top eigenspace, but also has several other interesting implications, e.g., top eigenspaces for different models have surprisingly high overlap, and top eigenvectors form low rank matrices when they are reshaped into the same shape as the corresponding weight matrix. We also empirically showed that the conjecture and its implications are true for both shallow and deep neural networks. Furthermore, we showed that our conjecture can be used to tighten the existing non-vacuous generalization bounds.

1.2.2 Learning-to-Learn for Tuning the Step Size

Training a neural network is a complicated process, and the parameters of the optimization algorithms have a big impact on the performances of the networks. These parameters are called hyperparameters. Examples of hyperparameters include step size, momentum, weight decay, etc. An improper choice of hyperparameters can cause many problems, e.g., overfitting, slow convergence, unstable training, or even divergence in training. Choosing proper hyperparameters is usually time-consuming, resource-expensive, and involves much human effort. To better select hyperparameters, people have developed a learning-to-learn approach (Andrychowicz et al., 2016), which views hyperparameter selection as a machine learning problem and learns these hyperparameters automatically. Although effective under many circumstances, learning-to-learn as a higher-level learning process is more difficult than regular training and suffers from multiple problems. For instance, the learning-to-learn problem is unstable in training, and the learned hyperparameters may not generalize well to other tasks (Metz et al., 2019).

We in Chapter 5 focus on a specific learning-to-learn problem: tuning the step size for a quadratic objective by running meta-gradient descent on a meta-objective. Although the inner objective is simple, this learning-to-learn problem is still difficult and has similar problems as practical circumstances. Under this framework, we study the meta-training instability problem in learning-to-learn and the generalization properties of the learned parameters. We show that the meta-gradient can explode or vanish if the meta-objective is not selected carefully, and computing the meta-gradient using backpropagation leads to numerical issues. Besides, we characterize the necessity of evaluating the hyperparameters on a separate validation set to guarantee their generalization performance. We prove that when the noise is large

and the sample size is not big enough, computing the meta-objective on a separate validation set has better generalization on unseen data. This is because computing the meta-objective on training data may not help us learn the correct parameter. We validate all our theoretical results in real datasets with practical neural networks.

1.3 Other Related Works

Theory for Self-Supervised Learning. There have been previous works that theoretically analyze different kinds of self-supervised learning methods and aim to explain why the representations learned in pre-training can be useful in downstream tasks. Most of the works in this area make assumptions about the data. Some works assume that the data are generated via explicit latent variable models and show that self-supervised learning methods can recover the latent variables. These models include hidden Markov model (Wei et al., 2021), topic models (Arora et al., 2016; Tosh et al., 2021a), graphical model (Zhang and Hashimoto, 2021), augmentation graph (HaoChen et al., 2021), Hierarchical Latent Tree model (Tian et al., 2020), etc. Another line of work assumes the data satisfy some properties, e.g., multi-view redundancy (Tosh et al., 2021c; Tsai et al., 2020) or conditional independence given the label (Lee et al., 2021; Saunshi et al., 2019), and prove that models can retrieve downstream-task-related information such as labels via self-supervised learning.

Besides the data perspective, some other works in this area focus on the properties of the model or representations. Wang and Isola (2020) showed that contrastive loss encourages alignment and uniformity of representations, Bansal et al. (2020) proved that a rational and robust-to-noise pre-trained model generalize well to downstream tasks, and Saunshi et al. (2020) reformulates the downstream task in the same form as pre-training and provides the performance guarantee.

Results for Global Neural Network Optimization Landscape. As for the optimization part, there has been a long line of work characterizing the global properties of neural network optimization landscape. Although the optimization landscape of a neural network is non-convex in general, people have shown that all local minima are global minima for some special neural networks, including linear neural networks (Kawaguchi, 2016; Hardt and Ma, 2016) and neural networks with structural or loss modifications (Liang et al., 2018; Kawaguchi and Kaelbling, 2020). However, this is not true in general, and spurious local minima are shown to be common in most neural networks (Safran and Shamir, 2018; Ding et al., 2019).

Despite the common existence of spurious local minima on neural network optimization landscape, another line of work shows that gradient descent usually converges to global minima, not to these spurious local minima or saddle points. This line of work includes Neural Tangent Kernel (Jacot et al., 2018; Allen-Zhu et al., 2019; Du et al., 2018) and mean-field theory (Mei et al., 2018; Chizat and Bach, 2018). However, the Neural Tangent Kernel behaves different from practical neural networks (Chizat and Bach, 1812; Arora et al., 2019), and the mean-field theory assumes the width of the neural network to be exponentially large and the depth to be two or three, which is unrealistic. Therefore, a satisfactory understanding of general neural network optimization is still lacking.

Theoretical Insights about Heuristics in Practical Neural Network Optimization. People have been using plenty of heuristics when optimizing neural networks empirically to help with the convergence and the generalization performance to new data. Some heuristics can work well in different domains, and some phenomena are commonly observed under different settings, so people believe these heuristics and phenomena

capture important properties of neural network optimization and perform theoretical analysis about them. Examples include batch normalization (Santurkar et al., 2018; Kohler et al., 2018; Arora et al., 2018b), sharpness of local minima (Dinh et al., 2017), Mixup (Guo et al., 2019; Carratino et al., 2020; Zhang et al., 2020, 2022a), and the Edge of Stability (Arora et al., 2022; Zhu et al., 2022). These theoretical studies improve our understanding of the heuristics or phenomena. They could also lead to new optimization algorithms that work better in practice. For instance, following the insight that flatter minima generalize better (Jiang et al., 2019; Keskar et al., 2017), Sharpness-Aware Minimization (Foret et al., 2020) leads the network parameters towards a flatter minima and usually generalizes better.

1.4 Bibliographic Notes

Chapter 2 is based on joint work (Chidambaram et al., 2023) with Muthu Chidambaram, Yu Cheng, and Rong Ge. Chapter 3 is based on joint work with Holden Lee and Rong Ge. Chapter 4 is based on joint work (Wu et al., 2020) with Yikai Wu, Xingyu Zhu, Annie N. Wang, and Rong Ge. Chapter 5 is based on joint work (Wang et al., 2021) with Xiang Wang, Shuai Yuan, and Rong Ge.

1.5 Future Directions

Our results about self-supervised representation learning rely on the generative models for the pre-training data. The two models used in this thesis are either simple or only opening the black box for one layer. A better analysis of the relationship between the learned representations and the structure of the pre-training data requires a model with more fine-grained structure and stronger expressive power. Another future direction is going beyond the generative model assumption, e.g., relying on the

connection between the pre-training task and the downstream task (Saunshi et al., 2020).

For the observations about neural network optimization, We only have rigorous results for one-layered or two-layered neural networks. Extending the existing results to deeper models or other phenomena is a meaningful future direction. Another future direction is to use our theoretical understanding to derive better algorithms for neural network optimization.

2

Masking Helps Sparse Coding

In this chapter, we present the benefit of self-supervised representation learning in sparse coding. We assume that the data are generated from a sparse linear combination of vectors in a dictionary and the goal of the pre-training task is to recover the dictionary. Under this setting, we compare two objectives for pre-training: the standard dictionary learning objective and the masking objective which is commonly used in self-supervised representation learning. We prove that when the data are noisy and the student model is over-realized, the masking objective can help successfully recover the dictionary while the standard objective fails. Therefore, for over-realized noisy sparse coding, masking can better capture the structure of input data during pre-training than the standard objective. This could explain why representations learned by self-supervised approaches yield better downstream performances.

2.1 Introduction

Modeling signals as sparse combinations of latent variables has been a fruitful approach in a variety of domains, and has been especially useful in areas such as medical

imaging (Zhang et al., 2017), neuroscience (Olshausen and Field, 2004), and genomics (Tibshirani and Wang, 2008), where learning parsimonious representations of data is of high importance. The particular case of modeling high-dimensional data in $\mathbb{R}^d$ as sparse *linear* combinations of a set of p vectors in $\mathbb{R}^d$ (referred to as a *dictionary*) has received significant attention over the past two decades, leading to the development of many successful algorithms and theoretical frameworks.

In this case, the typical assumption is that we are given data y_i generated as $y_i \sim Az_i + \epsilon_i$, where $A \in \mathbb{R}^{d \times p}$ is a ground-truth dictionary, z_i is a sparse vector, and ϵ_i is a noise vector. When the dictionary A is known a priori, the goal of modeling is to recover the sparse representations z_i, and the problem is referred to as *compressed sensing*. However, in many applications, we do not have access to the ground truth A, and instead hope to simultaneously learn a dictionary B that approximates A along with learning sparse representations of the data. This problem is referred to as *sparse coding* or *sparse dictionary learning*, which is what we focus on in this work.

One of the primary goals of analyses of sparse coding is to provide provable guarantees for recovering the ground-truth dictionary A, both with respect to specific algorithms and information-theoretically. Most prior work with such guarantees has focused exclusively on the setting where the learned dictionary B has the same size as the ground truth (in $\mathbb{R}^{d \times p}$), which is in line with the fact that recovery error is often formulated as some norm of $(B - A)$.

Unfortunately, in practice one does not necessarily have access to the structure of A. It is thus natural to consider what will happen (and how to formulate recovery error) if we learn a dictionary $B \in \mathbb{R}^{d \times p'}$ with $p' > p$, where it is possible to recover A as a sub-dictionary of B. The study of this *over-realized* setting was recently taken up in the work of Sulam et al. (2020), in which the authors showed that a modest

level of over-realization can be empirically and theoretically beneficial. However, the results of Sulam et al. (2020) are restricted to the noise-less setting where data is generated simply as $y_i \sim Az_i$, which motivates the following questions:

> *Does over-realized sparse coding run into pitfalls when there is noise in the data-generating process? And if so, is it possible to prevent this by designing new sparse coding algorithms?*

2.1.1 Main Contributions and Outline

In this work, we answer both of these questions in the affirmative. After providing the necessary background on sparse coding in Section 2.2, we show in Theorem 2.3.2 of Section 2.3 that, using standard sparse coding algorithms for learning over-realized dictionaries in the presence of noise leads to overfitting. In fact, our result shows that even if we allow the algorithms to access infinitely many samples and solve NP-hard problems, the learned dictionary B can still fail to recover A.

The key idea behind this result is that existing approaches for sparse coding largely rely on a two-step procedure (outlined in Algorithm 0): solving the compressed sensing problem $B\widehat{z} = y_i$ for a learned dictionary B, and then updating B based on a reconstruction objective $\|y_i - B\widehat{z}\|^2$. However, because we force $\widehat{z}$ to be sparse, by choosing B to have columns that are linear combinations of the columns of A, one can effectively get around the sparsity constraint on $\widehat{z}$. Consequently, it can be optimal to use such a B for reconstructing the data y_i, in which case A cannot be recovered as a sub-dictionary of B.

On the other hand, we show in Theorem 2.3.6 that for a large class of data-generating processes, it is possible to prevent this kind of B by *masking* data (as outlined in Algorithm 0): performing the compressed sensing step on a subset M of

the coordinates of y_i, and then computing the reconstruction loss on the complement of M. This idea of masking has seen great success in self-supervised learning (Devlin et al., 2019). Our result shows that it can lead to provable benefits in the context of sparse coding.

Finally, in Section 2.4 we conduct experiments comparing our masking approach with the standard sparse coding approach across several parameter regimes. In all of our experiments, we find that our masking approach leads to better ground truth recovery, which becomes more pronounced as the amount of over-realization increases.

2.1.2 Related Work

Compressed Sensing. The seminal works of Candes et al. (2006), Candes and Tao (2006), and Donoho (2006) established conditions on the dictionary $A \in \mathbb{R}^{d \times p}$, even in the case where $p \gg d$ (the *overcomplete* case), under which it is possible to recover (approximately and exactly) the sparse representations z_i from $Az_i + \epsilon_i$. In accordance with these results, several efficient algorithms based on convex programming (Tropp, 2006; Yin et al., 2008), greedy approaches (Tropp and Gilbert, 2007; Donoho et al., 2006; Efron et al., 2004), iterative thresholding (Daubechies et al., 2003; Maleki and Donoho, 2010), and approximate message passing (Donoho et al., 2009; Musa et al., 2018) have been developed for solving the compressed sensing problem. For comprehensive reviews on the theory and applications of compressed sensing, we refer the reader to the works of Candes and Wakin (2008) and Duarte and Eldar (2011).

Sparse Coding. Different framings of the sparse coding problem exist in the literature (Krause and Cevher, 2010; Bach et al., 2008; Zhou et al., 2009), but the

canonical formulation involves solving a non-convex optimization problem. Despite this hurdle, a number of algorithms (Engan et al., 1999; Aharon et al., 2006a; Mairal et al., 2010; Arora et al., 2013, 2014, 2015) have been established to (approximately) solve the sparse coding problem under varying conditions, dating back at least to the groundbreaking work of Olshausen and Field (1997) in computational neuroscience. A summary of convergence results and the conditions required on the data-generating process for several of these algorithms may be found in Table 1 of Gribonval et al. (2014).

In addition to algorithm-specific analyses, there also exists a complementary line of work on characterizing the optimization landscape of dictionary learning. This type of analysis is carried out by Gribonval et al. (2014) in the general setting of an overcomplete dictionary and noisy measurements with possible outliers, extending the previous line of work of Aharon et al. (2006b), Gribonval and Schnass (2010), and Geng et al. (2011).

However, as mentioned earlier, these theoretical results rely on learning dictionaries that are the same size as the ground truth. To the best of our knowledge, the over-realized case has only been studied by Sulam et al. (2020), and our work is the first to analyze over-realized sparse coding in the presence of noise.

Self-Supervised Learning. Training models to predict masked out portions of the input data is an approach to self-supervised learning that has led to strong empirical results in the deep learning literature (Devlin et al., 2019; Yang et al., 2019; Brown et al., 2020a; He et al., 2022). This success has spurred several theoretical studies analyzing how and why different self-supervised tasks can be used to improve model training (Tsai et al., 2020; Lee et al., 2021; Tosh et al., 2021b). The most closely related works to our own in this regard have studied the use of masking

objectives in autoencoders (Cao et al., 2022; Pan et al., 2022) and hidden Markov models (Wei et al., 2021).

2.2 Preliminaries and Setup

We first introduce some notation that we will use throughout the paper.

Notation. Given $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, 2, ..., n\}$. For a vector x, we write $\|x\|$ for the $\mathcal{L}_2$-norm of x and $\|x\|_0$ for the number of non-zero elements in x. We say a vector x is k-sparse if $\|x\|_0 \leqslant k$ and we use $\text{supp}\,(x)$ to denote the support of x. For a vector $x \in \mathbb{R}^d$ and a set $S \subseteq [d]$, we use $[x]_S \in \mathbb{R}^{|S|}$ to denote the restriction of x to those coordinates in S.

For a matrix A, we use A_i to denote the i-th column of A. We write $\|A\|_F$ for the Frobenius norm of A, and $\|A\|_{op}$ for the operator norm of A, and we write $\sigma_{\min}(A)$ and $\sigma_{\max}(A)$ for the minimum and maximum singular values of A. For a matrix $A \in \mathbb{R}^{d \times q}$ and $S \subseteq [q]$, we use $A_S \in \mathbb{R}^{d \times |S|}$ to refer to A restricted to the columns whose indices are in S. We use I_d to denote the $d \times d$ identity matrix. Finally, for $M \subseteq [d]$, we use $P_M \in \mathbb{R}^{|M| \times d}$ to refer to the matrix whose action on x is $P_M x = [x]_M$. Note that for a $d \times q$ matrix A, $P_M A$ would give a subset of rows of A, which is different from the earlier notation A_S which gives a subset of columns.

2.2.1 Background on Sparse Coding

We consider the sparse coding problem in which we are given measurements $y \in \mathbb{R}^d$ generated as $Az + \epsilon$, where $A \in \mathbb{R}^{d \times p}$ is a ground-truth dictionary, $z \in \mathbb{R}^p$ is a k-sparse vector distributed according to a probability measure $\mathbb{P}_z$, and $\epsilon \in \mathbb{R}^d$ is a noise term with i.i.d. entries. The goal is to use the measurements y to reconstruct a dictionary B that is as close as possible to the ground-truth dictionary A.

In the case where B has the same dimensions as A, one may want to formulate this notion of "closeness" (or recovery error) as $\|A - B\|_F^2$. However, directly using the Frobenius norm of $(A-B)$ is too limited, as it is sufficient to recover the columns of A up to permutations and sign flips. Therefore, a common choice of recovery error (Gribonval et al., 2014; Arora et al., 2015) is the following:

$$\min_{P \in \Pi} \|A - BP\|_F^2 \tag{2.1}$$

where Π is the set of orthogonal matrices whose entries are 0 or ± 1.

In the over-realized setting, when $B \in \mathbb{R}^{d \times p'}$ with $p' > p$, Equation (2.1) no longer makes sense as A and B do not have the same size. In this case, one can generalize Equation (2.1) to measure the distance between each column of A and the column closest to it in B (up to change of sign). This notion of recovery was studied by Sulam et al. (2020), and we use the same formulation in this work:

$$d_R(A, B) \triangleq \frac{1}{p} \sum_{i=1}^{p} \min_{j \in [p'], c \in \{-1,1\}} \|A_i - cB_j\|^2 \tag{2.2}$$

Note that Equation (2.2) introduced the coefficient $1/p$ in the recovery error and thus corresponds to the *average* distance between A_i and its best approximation in B. Also, Equation (2.2) only allows sign changes, even though for reconstructing Az, it is sufficient to recover the columns of A up to arbitrary scaling. In our experiments we enforce A and B to have unit column norms so a sign change suffices; in theory one can always modify the B matrix to have correct norm so it also does not change our results.

Given access to only measurements y, the algorithm cannot directly minimize the recovery error $d_R(A, \cdot)$. Instead, sparse coding algorithms often seek to minimize the

following surrogate loss:

$$\ell(B) = \mathbb{E}_y \left[\min_{\widehat{z} \in \mathbb{R}^{p'}} \|y - B\widehat{z}\|^2 + h(\widehat{z}) \right] \tag{2.3}$$

where h is a sparsity-promoting penalty function. Typical choices of h include hard sparsity ($h(\widehat{z}) = 0$ if $\widehat{z}$ is k-sparse and $h(\widehat{z}) = \infty$ otherwise) as well as the $\mathcal{L}_1$ penalty $h(\widehat{z}) = \|\widehat{z}\|_1$. While hard sparsity is closer to the assumption on the data-generating process, it is well-known that optimizing under exact sparsity constraints is NP-hard in the general case (Natarajan, 1995). When $h(\widehat{z}) = \|\widehat{z}\|_1$ is used, the learning problem is also known as basis pursuit denoising (Chen and Donoho, 1994) or Lasso (Tibshirani, 1996).

Equation (2.3) is the population loss one wishes to minimize when learning a dictionary B. In practice, sparse coding algorithms must work with a finite number of measurements $y_1, y_2, \ldots, y_n$ obtained from the data-generating process and instead minimize the empirical loss $\widetilde{\ell}(B)$:

$$\widetilde{\ell}(B) = \frac{1}{n} \sum_{i=1}^{n} \min_{\widehat{z} \in \mathbb{R}^{p'}} \|y_i - B\widehat{z}\|^2 + h(\widehat{z}) \tag{2.4}$$

2.2.2 Sparse Coding via Orthogonal Matching Pursuit

Most existing approaches for optimizing Equation (2.4) can be categorized under the general alternating minimization approach described in Algorithm 0.

Algorithm 1 Alternating Minimization Framework

Input: Data y, Dictionary $B^{(t)} \in \mathbb{R}^{d \times p'}$
Decoding Step: Solve $B^{(t)}\widehat{z} = y$ for k-sparse $\widehat{z}$
Update Step: Update $B^{(t)}$ to $B^{(t+1)}$ by performing a gradient step on loss computed using $B^{(t)}\widehat{z}$ and y

At iteration t, Algorithm 0 performs a decoding/compressed sensing step using the current learned dictionary $B^{(t)}$ and the input data y. As mentioned in Section

2.1.2, there are several well-studied algorithms for this decoding step. Because we are interested in enforcing a hard-sparsity constraint, we restrict our attention to algorithms that are guaranteed to produce a k-sparse representation in the decoding step.

We thus focus on Orthogonal Matching Pursuit (OMP) (Mallat and Zhang, 1993; Rubinstein et al., 2008), which is a simple greedy algorithm for the decoding step. The basic procedure of OMP is to iteratively expand a subset $T \subset [p']$ of atoms (until $|T| = k$) by considering the correlation between the unselected atoms in the current dictionary $B^{(t)}$ and the residual $\left(y - B_T^{(t)} \arg\min_{\widehat{z} \in \mathbb{R}^{|T|}} \left\| y - B_T^{(t)} \widehat{z} \right\|^2\right)$ (i.e., the least squares solution using atoms in T). A more precise description of the algorithm can be found in Rubinstein et al. (2008). Moving forward, we will use $g_{\text{OMP}}(y, B, k)$ to denote the k-sparse vector $\widehat{z} \in \mathbb{R}^{p'}$ obtained by running the OMP algorithm on an input dictionary B and a measurement y.

2.2.3 Conditions on the Data-Generating Process

For the data-generating process $y \sim Az + \epsilon$, it is in general impossible to successfully perform the decoding step in Algorithm 0 even with access to the ground-truth dictionary A. As a result, several conditions have been identified in the literature under which it is possible to provide guarantees on the success of decoding the sparse representation z. We recall two of the most common ones (Candes and Tao, 2005).

Definition 2.2.1. [*Restricted Isometry Property (RIP)*] We say that a matrix $A \in \mathbb{R}^{d \times p}$ satisfies (s, δ_s)-RIP if the following holds for all s-sparse $x \in \mathbb{R}^p$:

$$(1 - \delta_s)\|x\|^2 \leqslant \|Ax\|^2 \leqslant (1 + \delta_s)\|x\|^2 \tag{2.5}$$

Definition 2.2.2. [*μ-Incoherence*] A matrix $A \in R^{d \times p}$ with unit norm columns is μ-incoherent if:

$$|\langle A_i, A_j \rangle| \leqslant \mu \quad \text{for all } i \neq j \tag{2.6}$$

These two properties are closely related. For example, as a consequence of the Gershgorin circle theorem, (δ_s/s)-incoherent matrices must satisfy (s, δ_s)-RIP.

Given the prominence of RIP and incoherence conditions in the compressed sensing and sparse coding literature, there has been a large body of work investigating families of matrices that satisfy these conditions. We refer the reader to Baraniuk et al. (2008) for an elegant proof that a wide class of random matrices in $\mathbb{R}^{d \times p}$ (i.e. subgaussian) satisfy (k, δ)-RIP with high probability depending on δ, k, p, and d. For an overview of deterministic constructions of such matrices, we refer the reader to Bandeira et al. (2012) and the references therein.

2.3 Main Results

Having established the necessary background, we now present our main results. Our first result shows that minimizing the population reconstruction loss with a hard-sparsity constraint can lead to learning a dictionary B that is far from the ground truth. We specifically work with the loss defined as:

$$L(B, k) = \mathbb{E}_y \left[\min_{\|\widehat{z}\|_0 \leqslant k} \|y - B\widehat{z}\|^2 \right] \tag{2.7}$$

Note that in the definition of $L(B, k)$, we are considering an NP-hard optimization problem (exhaustively searching over all k-sparse supports). We could instead replace this exhaustive optimization with an alternative least-squares-based approach (so long as it is better than random), and our proof techniques for Theorem 2.3.2 would

still work. We consider this version only to simplify the presentation.

We now show that, under appropriate settings, there exists a dictionary B whose population loss $L(B, k)$ is smaller than that of A, while $d_R(A, B)$ is bounded away from 0 by a term related to the noise in the data-generating process.

Assumption 2.3.1. Let $A \in \mathbb{R}^{d \times p}$ be an arbitrary matrix with unit-norm columns satisfying $(2k, \delta)$-RIP for $k = o\left(\frac{d}{\log p}\right)$ and $\delta = o(1)$, and suppose $\sigma^2_{\min}(A) = \Omega(p/d)$. We assume each measurement y is generated as $y \sim Az + \epsilon$, where z is a random vector drawn from an arbitrary probability measure $\mathbb{P}_z$ on k-sparse vectors in $\mathbb{R}^p$, and $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ for some $\sigma > 0$.

Theorem 2.3.2. *[Overfitting to Reconstruction Loss] Consider the data-generating model in Assumption 2.3.1 and define $\Lambda(z)$ to be:*

$$\Lambda(z) = \inf\{t \mid \mathbb{P}_z(z \geqslant t) \leqslant 1/d\}. \tag{2.8}$$

Then for $q = \Omega(p^2 \max(d\sigma^2, \Lambda(z)^2)/\sigma^2)$, there exists a $B \in R^{d \times q}$ such that $L(B, k) \leqslant L(A, k) - \Omega(k\sigma^2)$ and $d_R(A, B) = \Omega(\sigma^2)$.

Proof Sketch. The key idea is to first determine how much the loss can be decreased by expanding from k-sparse combinations of the columns of A to $2k$-sparse combinations, i.e., lower bound the gap between $L(A, k)$ and $L(A, 2k)$. After this, we can construct a dictionary B whose columns form an ϵ-net (with $\epsilon = \sigma^2$) for all 2-sparse combinations of columns of A. Any $2k$-sparse combination of columns in A can then be approximated as a k-sparse combination of columns in B, which is sufficient for proving the theorem.

Remark 2.3.3. Before we discuss the implications of Theorem 2.3.2, we first verify that Assumption 2.3.1 is not vacuous, and in fact applies to many matrices of interest.

This follows from a result of Rudelson and Vershynin (2009), which shows that after appropriate rescaling, rectangular matrices with i.i.d. subgaussian entries satisfy the singular value condition in Assumption 2.3.1. Furthermore, such matrices will also satisfy the RIP condition so long as k is not too large relative to d and p, as per Baraniuk et al. (2008) as discussed in the previous section.

Theorem 2.3.2 shows that learning an appropriately over-realized dictionary fails to recover the ground truth *independent* of the distribution of z. This means that even if we let the norm of the signal Az in the data-generating process be arbitrarily large, with sufficient over-realization we may still fail to recover the ground-truth dictionary A by minimizing $L(B, k)$.

We also observe that the amount of over-realization necessary in Theorem 2.3.2 depends on how well $z \sim \mathbb{P}_z$ can be bounded with reasonably high probability. If z is almost surely bounded (as is frequently assumed), we can obtain the following cleaner corollary of Theorem 2.3.2.

Corollary 2.3.4. *Consider the same settings as Theorem 2.3.2 with the added stipulation that* $\mathbb{P}_z(\|z\| \leqslant C) = 1$ *for a universal constant* C*. Then for* $q = \Omega(p^2 d)$*, there exists a* $B \in R^{d \times q}$ *such that* $d_R(A, B) = \Omega(\sigma^2)$ *and* $L(B, k) \leqslant L(A, k) - \Omega(k\sigma^2)$*.*

The reason that we can obtain a smaller population loss than the ground truth in Theorem 2.3.2 is because we can make use of the extra capacity in B to overfit the noise ϵ in the data-generating process. To prevent this, our key idea is to perform the decoding step $B\widehat{z} = y$ on a subset of the dimensions of y - which we refer to as the *unmasked* part of y - and then evaluate the loss of B using the complement of that subset (the *masked* part of y). Intuitively, because each coordinate of the noise ϵ is independent, a dictionary B that well-approximates the noise in the unmasked part of y will have no benefit in approximating the noise in the masked part of y.

We can formalize this as the following masking objective:

$$L_{mask}(B, k, M) = \mathbb{E}_y \left[\left\| [y]_{[d]\setminus M} - [B\widehat{z}]_{[d]\setminus M} \right\|^2 \right] \tag{2.9}$$

$$\text{where } \widehat{z}([y]_M) = g_{\text{OMP}}(y, B, k) \tag{2.10}$$

In defining L_{mask}, we have opted to use g_{OMP} in the inner minimization step, as opposed to the exhaustive arg min in the definition of L. Similar to the discussion earlier, we could have instead used any other approach based on least squares to decode $\widehat{z}$ (including the exhaustive approach), so long as we have guarantees on the probability of failing to recover the true code z given access to the ground-truth dictionary A. This choice of using OMP is mostly to tie our theory more closely with our experiments.

Now we present our second main result which shows, in contrast to Theorem 2.3.2, that optimizing L_{mask} prevents overfitting noise (albeit in a different but closely related setting).

Assumption 2.3.5. Let $A \in \mathbb{R}^{d\times p}$ be an arbitrary matrix such that there exists an $M \subset [d]$ with $P_M A$ being μ-incoherent with $\mu \leqslant C/(2k-1)$ for a universal constant $C < 1$. We assume each measurement y is generated as $y \sim Az + \epsilon$, where $[z]_{\text{supp}(z)} \sim \mathcal{N}(0, \sigma_z^2 \mathrm{I}_k)$ with $\text{supp}(z)$ drawn from an arbitrary probability distribution over all size-k subsets of $[d]$, and $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ for some $\sigma > 0$.

Theorem 2.3.6. *[Benefits of Masking] Consider the data-generating model in Assumption 2.3.5. For any non-empty mask $M \subset [d]$ such that $P_M A$ satisfies the μ-incoherence condition in the assumption, we have*

$$\lim_{\sigma_z \to \infty} \left(L_{mask}(A, k, M) - \min_B L_{mask}(B, k, M) \right) = 0 \tag{2.11}$$

That is, as the expected norm of the signal Az increases, there exist minimizers B of L_{mask} such that $d_R(A, B) \to 0$.

Proof Sketch. The proof proceeds by expanding out $L_{mask}(B, k, M)$ and using the fact that $[B\widehat{z}^*]_{[d]\setminus M}$ is independent of $[\epsilon]_{[d]\setminus M}$ to obtain a quantity that closely resembles the prediction risk considered in analyses of linear regression. From there we show that the Bayes risk is lower bounded by the risk of a regularized least squares solution with access to a support oracle. We then rely on a result of Cai and Wang (2011) to show that $g_{\mathrm{OMP}}([y]_M)$ recovers the support of z with increasing probability as $\sigma_z \to \infty$, and hence its risk converges to the aforementioned prediction risk.

Remark 2.3.7. As before, so long as the mask M is not too small, matrices with i.i.d. subgaussian entries will satisfy the assumptions on A in Assumption 2.3.5. In particular, the set of ground truth dictionaries satisfying Assumptions 2.3.1 *and* 2.3.5 is non-empty.

Comparing Theorem 2.3.6 to Theorem 2.3.2, we see that approximate minimizers of L_{mask} can achieve arbitrarily small recovery error, so long as the signal Az is large enough; whereas for L, there always exist minimizers whose recovery error is bounded away from 0. We note that having the expected norm of the signal be large is effectively necessary to hope for recovering the ground truth in our setting, as in the presence of Gaussian noise there is always some non-zero probability that the decoding step can fail. Full proofs of Theorems 2.3.2 and 2.3.6 can be found in Section A.1 of the Appendix.

2.4 Experiments

In this section, we examine whether the separation between the performance of sparse coding with or without masking (demonstrated by Theorems 2.3.2 and 2.3.6) mani-

fests in practice. To do so, we need to make a few concessions from the theoretical settings introduced in Sections 2.2.1 and 2.2.3. Firstly, we cannot directly optimize the expectations in L and L_{mask} as defined in Equations (2.7) and (2.9), so we instead optimize the corresponding empirical versions defined in the same vein as Equation (2.4). Another issue is that the standard objective L requires solving the optimization problem $\min_{\|\widehat{z}\|_0 \leqslant k} \|y - B\widehat{z}\|^2$, which is NP-hard in general. In order to experiment with reasonably large values of $d, p,$ and p' and to be consistent with the decoding step in L_{mask}, we thus approximately solve the aforementioned optimization problem using OMP.

Algorithm 2 Algorithm for Optimizing L

Input: Data $\{y_1, ..., y_T\}$, Dictionary $B^{(0)} \in \mathbb{R}^{d \times p'}$, Learning Rate $\eta \in \mathbb{R}^+$
for $t = 0$ **to** $T - 1$ **do**
  $z \leftarrow g_{\text{OMP}}(y_{t+1}, B^{(t)})$
  $B^{(t+1)} \leftarrow B^{(t)} - \eta \nabla_{B^{(t)}} \left\| y_{t+1} - B^{(t)} z \right\|^2$
  $B^{(t+1)} \leftarrow \text{Proj}_{\mathbb{S}^{d-1}} B^{(t+1)}$
end for

Algorithm 3 Algorithm for Optimizing L_{mask}

Input: Data $\{y_1, ..., y_T\}$, Dictionary $B^{(0)} \in \mathbb{R}^{d \times p'}$, Learning Rate $\eta \in \mathbb{R}^+$, Mask Size $m \in [d]$
for $t = 0$ **to** $T - 1$ **do**
  $M \leftarrow$ Uniformly random subset of size m from $[d]$
  $z \leftarrow g_{\text{OMP}}([y_{t+1}]_M, B^{(t)})$
  $B^{(t+1)} \leftarrow B^{(t)} - \eta \nabla_{B^{(t)}} \left\| [y_{t+1}]_{M^c} - [B^{(t)} z]_{M^c} \right\|^2$
  $B^{(t+1)} \leftarrow \text{Proj}_{\mathbb{S}^{d-1}} B^{(t+1)}$
end for

The approaches for optimizing L and L_{mask} given n samples from the data-generating process are laid out in Algorithms 0 and 0, in which we use $\text{Proj}_{\mathbb{S}^{d-1}} B$ to denote the result of normalizing all of the columns of B. We also use M^c as a shorthand in Algorithm 0 to denote $[d] \backslash M$.

We point out that Algorithm 0 introduces some features that were not present in the theory of the masking objective; namely, in each iteration, we randomly sample a new mask of a pre-fixed size. This is because if we were to run gradient descent using a single, fixed mask M at each iteration, as we don't differentiate through the OMP steps, the gradient with respect to $B^{(t)}$ computed on the error $\left\|[y]_{[d]\setminus M} - [Bz]_{[d]\setminus M}\right\|^2$ would be non-zero only for those rows of B corresponding to the indices $[d]\setminus M$. To avoid this issue, we sample new masks in each iteration so that each entry of B can be updated. There are alternative approaches that can achieve this; i.e. deterministically cycling through different masks, but we found them to have similar performance.

While we will analyze the performance of Algorithms 0 and 0 across several different experimental setups over the next few subsections, we describe the following facets shared across all setups. We generate a dataset of $n = 1000$ samples $y_i = Az_i + \epsilon_i$, where $A \in \mathbb{R}^{d\times p}$ is a standard Gaussian ensemble with normalized columns, the z_i have uniformly random k-sparse supports whose entries are i.i.d. $\mathcal{N}(0, 1)$, and the ϵ_i are mean zero Gaussian noise with some fixed variance (which we will vary in our experiments). We also normalize the z_i so as to constrain ourselves to the bounded-norm setting of Corollary 2.3.4. In addition to the 1000 samples constituting the dataset, we also assume access to a held-out set of p' samples from the data-generating process for initializing the dictionary $B^{(0)} \in \mathbb{R}^{d\times p'}$.

For training, we use batch versions of Algorithms 0 and 0 in which we perform gradient updates with respect to the mean losses computed over $\{y_1, ..., y_B\}$ with $B = 200$ as the batch size. For the actual gradient step, we use Adam (Kingma and Ba, 2014) with its default hyperparameters of $\beta_1 = 0.9, \beta_2 = 0.999$ and a learning rate of $\eta = 0.001$, as we found Adam trains *significantly* faster than SGD (and we

ran into problems when using large learning rates for SGD). We train for 500 epochs (passes over the entire dataset) for both Algorithms 0 and 0. For Algorithm 0, we always use a mask size of $d - \lfloor d/10 \rfloor$, which we selected based off of early experiments. We ensured that, even for this fairly large mask size, the gradient norms for both L and L_{mask} were of the same order in our experiments and that 500 epochs were sufficient for training.

We did not perform extensive hyperparameter tuning, but we found that the aforementioned settings performed better than the alternative choices we tested for both algorithms across all experimental setups. Our implementation is in PyTorch (Paszke et al., 2019a), and all of our experiments were conducted on a single P100 GPU.

2.4.1 Scaling Over-Realization

We first explore how the choice of p' for the learned dictionary B affects the empirical performance of Algorithms 0 and 0 when the other parameters of the problem remain fixed. Theorem 2.3.2 and Corollary 2.3.4 indicate that the performance of Algorithm 0 should suffer as we scale p' relative to d and p.

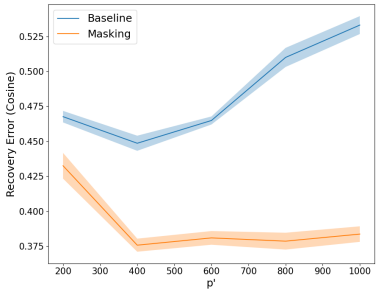
In order to test whether this is actually the case in practice, we consider samples generated as described above with $A \in \mathbb{R}^{d \times p}$ for $d = 100, p = 200,$ and $\|z\|_0 = k = 5$ fixed, while scaling the number of atoms p' in B from $p' = p$ (exactly realized) to $p' = n$ (over-realized and overparameterized). We choose $\epsilon_i \sim \mathcal{N}(0, 1/d)$, which is a high noise regime as the expected norm of the noise ϵ_i will be comparable to that of the signal Az_i. To make it computationally feasible to run several trials of our experiments, we consider the p' values $\{200, 400, 600, 800, 1000\}$ and do not consider more fine-grained interpolation between p and n.

For the training process, we consider two different initialization of $B^{(0)}$. In the first case, we initialize $B^{(0)}$ to have columns corresponding to the aforementioned set of held-out p' (normalized) samples from the data-generating process, as this is a standard initialization choice that has been known to work well in practice (Arora et al., 2015). However, this initialization choice in effect corresponds to a dataset of $n + p'$ samples, and it is fair to ask whether this initialization benefit is worth the sample cost relative to a random initialization. Our initial experiments showed that this was indeed the case, i.e. random initialization with access to p' additional samples did not perform better, so we focus on this sample-based initialization. That being said, we did not find the ordering of the performances of Algorithms 0 and 0 sensitive to the initializations we considered, only the final absolute performance in terms of $d_R(A, B)$.

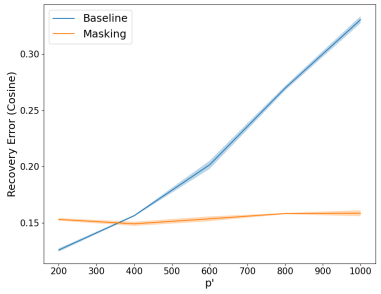
In addition to this purely sample-based initialization, we also consider a "local" initialization of $B^{(0)}$ to the ground truth A itself concatenated with $p' - p$ normalized samples. This is obviously not intended to be a practical initialization; the goal here is rather to analyze the extent of overfitting to the noise ϵ_i in the dataset for both algorithms. Namely, we expect that Algorithm 0 will move further away from the ground truth than Algorithm 0.

The results for training using these initializations for both algorithms and then computing the final dictionary recovery errors $d_R(A, B)$ are shown in Figure 2.1. We use cosine distance when reporting the error $d_R(A, B)$ since the learned dictionary B also has normalized columns, so Euclidean distance only changes the scale of the error curves and not their shapes.

For both choices of initialization, we observe that Algorithm 0 outperforms Algorithm 0 as p' increases, with this gap only becoming more prominent for larger



(a) Training from sample initialization



(b) Training from local initialization

FIGURE 2.1: Comparison of Algorithm 0 (Baseline) and Algorithm 0 (Masking) under the settings of Sections 2.4.1. Each curve represents the mean of 5 training runs, with the surrounded shaded area representing one standard deviation.

p'. Furthermore, we find that recovery error actually *worsens* for Algorithm 0 for every choice of $p' > p$ for both initializations in our setting. While this is possibly unsurprising for initializing at the ground truth, it is surprising for the sample-based initialization which does not start at a low recovery error. On the other hand, training using Algorithm 0 improves the recovery error from initialization when using sample-based initialization for every choice of p' except $p' = n$, which again corresponds to the overparameterized regime in which it is theoretically possible to memorize every sample as an atom of B.

Additionally, we also see that the performance of Algorithm 0 is much less sensitive to the level of over-realization in B. When training from local initialization, Algorithm 0 retains a near-constant level of error/overfitting as we scale p'. Similarly, when training from sample initialization, performance does not degrade as we scale p', and in fact improves initially with a modest level of over-realization.

This improvement up to a certain amount of over-realization (in our case $p' = 2p$) is seen even in the performance of Algorithm 0 for sample initialization (although note that while the recovery error is better for $p' = 2p$ compared to $p' = p$, training still

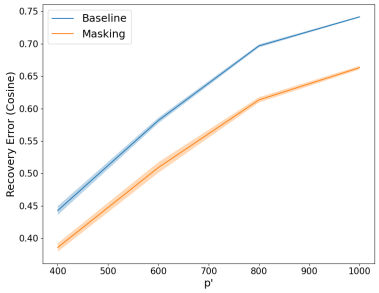
makes the error worse than initialization for Algorithm 0). A similar phenomenon was observed in Sulam et al. (2020) in the setting where $y_i = Az_i$ (no noise), and we find it interesting that the phenomenon is (seemingly) preserved even in the presence of noise. We do not investigate the optimal level of over-realization any further, but believe it would be a fruitful direction for future work.

2.4.2 Scaling All Parameters

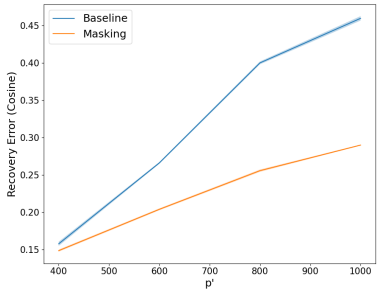
The experiments of Section 2.4.1 illustrate that for the fixed choices of d, p, and k that we used, scaling the over-realization of B leads to rapid overfitting in the case of Algorithm 0, while Algorithm 0 maintains good performance. To verify that this is not an artifact of the choices of $d, p,$ and k that we made, we also explore what happens when over-realization is kept at a fixed ratio to the other setting parameters while they are scaled.

For these experiments, we consider $A \in \mathbb{R}^{d \times p}$ for $d \in \{100, 150, 200, 250\}$ and scale p as $p = 2d$ and k as $k = \lfloor d/20 \rfloor$ to (approximately) preserve the ratio of atoms and sparsity to dimension from the previous subsection. We choose to scale p' as $p' = 2p$ since that was the best-performing setting (for the baseline) from the experiments of Figure 2.1. We keep the noise variance at $1/d$ to stay in the relatively high noise regime.

As before, we consider a sample-based initialization as well as a local initialization near the ground truth dictionary A. The results for both Algorithms 0 and 0 under the described parameter scaling are shown in Figure 2.2. Once again we find that Algorithm 0 has superior recovery error, with this gap mostly widening as the parameters are scaled. However, unlike the case of fixed $d, p,$ and k, this time the performance of Algorithm 0 also degrades with the scaling. This is to be expected,



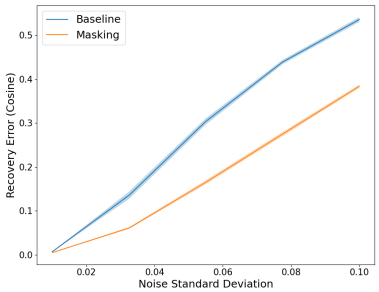
(a) Training from sample initialization



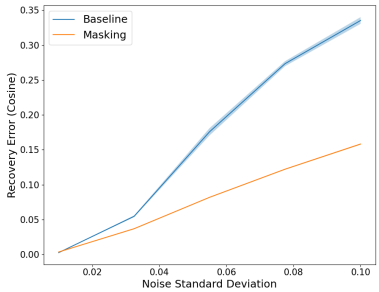
(b) Training from local initialization

FIGURE 2.2: Comparison of Algorithm 0 (Baseline) and Algorithm 0 (Masking) under the settings of Section 2.4.2 (all parameter scaling).

as increasing p leads to more ground truth atoms that need to be recovered well in order to have small $d_R(A, B)$.



(a) Training from sample initialization



(b) Training from local initialization

FIGURE 2.3: Comparison of Algorithm 0 (Baseline) and Algorithm 0 (Masking) under the settings of Section 2.4.3 (noise scaling).

2.4.3 Analyzing Different Noise Levels

The performance gaps shown in the plots of Figures 2.1 and 2.2 are in the high noise regime, and thus it is fair to ask whether (and to what extent) these gaps are preserved at lower noise settings. We thus revisit the settings of Section 2.4.1 (choosing $d, p,$ and k to be the same) and fix $p' = 1000$ (the maximum over-realization we consider). We then vary the variance of the noise ϵ_i from $1/d^2$ to $1/d$ linearly, which corre-

sponds to the standard deviations of the noise being $\{0.01, 0.0325, 0.055, 0.0775, 0.1\}$.

Results are shown for the sample-based initialization as well as the local initialization in Figure 2.3. Here we see that when the noise variance is very low, there is virtually no difference in performance between Algorithms 0 and 0. Indeed, when the variance is $1/d^2$ we observe that both algorithms are able to near-perfectly recover the ground truth, even from the sample-based initialization.

However, as we scale the noise variance, the gap between the performance of the two algorithms resembles the behavior seen in the experiments of Sections 2.4.1 and 2.4.2.

2.5 Conclusion

In summary, we have shown in Sections 2.3 and 2.4 that applying the standard frameworks for sparse coding to the case of learning over-realized dictionaries can lead to overfitting the noise in the data. In contrast, we have also shown that by carefully separating the data used for the decoding and update steps in Algorithm 0 via masking, it is possible to alleviate this overfitting problem both theoretically and practically. Furthermore, the experiments of Section 2.4.3 demonstrate that these improvements obtained from masking are not at the cost of worse performance in the low noise regime, indicating that a practitioner may possibly use Algorithm 0 as a drop-in replacement for Algorithm 0 when doing sparse coding.

Our results also raise several questions for exploration in future work. Firstly, in both Theorem 2.3.6 and our experiments we have constrained ourselves to the case of sparse signals that follow Gaussian distributions. It is natural to ask to what extent this is necessary, and whether our results can be extended (both theoretically and empirically) to more general settings (we expect, at the very least, that parts of

Assumptions 2.3.1 and 2.3.5 can be relaxed). Additionally, we have focused on sparse coding under hard-sparsity constraints and using orthogonal matching pursuit, and it would be interesting to study whether our ideas can be used in other sparse coding settings.

Beyond these immediate considerations, however, the intent of our work has been to show that there is still likely much to be gained from applying ideas from recent developments in areas such as self-supervised learning to problems of a more classical nature such as sparse coding. Our work has only touched on the use of a single such idea (masking), and we hope that future work looks into how other recently popular ideas can potentially improve older algorithms.

Finally, we note that this work has been mostly theoretical in nature, and as such do not anticipate any direct misuses or negative impacts of the results.

3

What Downstream Tasks Provably Benefit from Pre-Training?

In the previous chapter, we have proved the benefit of self-supervised learning during pre-training when the data are generated from a sparse coding model, which is a simple and well-studied generative model. In this chapter, we are going to analyze the benefit of self-supervised pre-training for more complicated data. We only require that each entry of the input sequence is generated by a log-linear model given the representation of its context, and this representation can be an arbitrarily complicated function of the input. Under this assumption, we investigate what kinds of downstream tasks can benefit from the pre-training process. When the pre-training task is language modeling and the downstream task is binary sequence classification depending on a simple function of the hidden representation, we found two necessary conditions and one sufficient condition for provably guaranteeing the model's pre-training performance to transfer to its downstream performance. The two necessary conditions include robustness to very small probabilities and structure to handle the shift-invariance of the softmax function. The sufficient condition is the existence of

an "anchor vector", which we have also empirically verified in various large language models.

3.1 Introduction

Large-scale pre-trained language models have achieved strong performance in a wide range of downstream tasks, including natural language inference (Devlin et al., 2018), reading comprehension (Brown et al., 2020b), etc. For many of these tasks, training a linear classifier on top of the hidden-layer representations generated by the pre-trained models can already provide near state-of-the-art results (Belinkov et al., 2017). Despite some empirical investigations about the zero-shot applications of these pre-trained models, there is little theoretical understanding about their empirical success. In this paper, we aim to theoretically investigate this core question:

> *When can the* ***representations*** *from pre-trained models transfer to downstream tasks that are* ***very different*** *from pre-training?*

This is a fundamental question in understanding why good performance in pre-training generalizes to give good performance on downstream tasks. Unlike the notion of generalization in traditional learning theory where the models are evaluated in the same task and the test data are sampled from the same distribution as the training data, here the downstream tasks can be very different from training. For instance, people can train large language models using cross-entropy loss on a language modeling task with webpage data, and evaluate the models using classification accuracy on text classification in news articles. This difference between pre-training and downstream tasks makes it challenging to theoretically understand the success of these language models, and previous results about generalization cannot be directly applied here.

To overcome this challenge, we need a way to model the relationship between the pre-training and downstream tasks. Previous research has taken several approaches in this direction: Wei et al. (2021) assumes a latent-variable generative model for the data and that the downstream task depends on the latent variables; Nikunj et al. (2021) formulates the downstream classification task as a language modeling task which is similar to the pre-training task. These works either rely on strong explicit assumptions about structure in the data or treat the pre-trained model entirely as a black box.

3.1.1 Our Contributions

In this paper, we consider a very general model for the data and open the black box of the pre-trained model at the last layer. Specifically, for an input sequence $x = (x_1, \cdots, x_L)$ where the entries comes from a dictionary $\{1, \cdots, n\}$, we assume the observation probability of x_i satisfies

$$p^*(x_i = j | x_{-i}) \propto \exp\big(\langle v^*_{-i}(x_{-i}), v^*_j \rangle\big),$$

where x_{-i} is the sequence x without x_i, v^*_j is a vector only depending on word j, and v^*_{-i} is an arbitrary function. Our model does not put any constraint on the functions v^*_{-i}, so it can be very complicated, such as BERT (Devlin et al., 2018) or GPT-3 (Brown et al., 2020b). We also allow the distribution of sequences x to be different in pre-training and downstream tasks. This makes our setting more general than previous latent models Wei et al. (2021); Arora et al. (2016).

We assume the pre-training task is predicting a word from its context, and the downstream task is a binary sequence classification, e.g., sentiment classification. During pre-training, for every input sequence x, we want our model to predict the

"label" x_i from x_{-i}. We use a log-linear model to learn the ground-truth probability: $p(x_i = j|x_{-i}) \propto \exp(\langle v_{-i}(x_{-i}), v_j \rangle)$. We define $z_j^* := \langle v_{-i}^*(x_{-i}), v_j^* \rangle$ and call $z^* := (z_1^*, \cdots, z_n^*)$ the logits. We also assume that the downstream task only depends on a ground-truth function of the logits $f^*(z^*)$. In reality, we only have access to $z := (\langle v_{-i}(x_{-i}), v_j \rangle)_{j=1}^n$ and will learn a function $f(z)$ for the downstream task. More detail about this model will be provided in Section 3.2.

We focus on the representations from the model, so to simplify the problem, we assume that the student model can be optimized to achieve a small KL divergence with the true word probabilities during training. This is often the case in practice.

Under this setting, the question we ask above becomes:

> *If the downstream task depends on a simple function of the logits $f^*(z^*)$ and we have access to a student model p such that $\mathbb{E}_x[D_{\mathrm{KL}}(p^*(x_i|x_{-i})||p(x_i|x_{-i}))]$ is small, under what condition would there be a function f such that $f(z) \approx f^*(z^*)$?*

We answer this question by providing two necessary conditions and one sufficient condition for f^* to be provably learned by the student model. The necessary conditions include insensitivity to super-small probability words and the structure in representations to handle the shift-invariance of the softmax function.

As for the sufficient condition, we proposed and empirically verified the existence of an "anchor vector" that can tackle the shift-invariance of softmax. We also assume that f^* only depends on a small set of words, which is usually achieved by adding proper prompts to the input sequence. For instance, appending "This movie is" to movie reviews will make the model focus on indicative adjectives like "great" or "bad", and adding "This article is about" concentrates the probabilities on words for categories like "sports" or "science" (Nikunj et al., 2021). Under this setting, we

theoretically proved that if f^* is a one-hidden-layer ReLU network of the logits z^*, the existence of the "anchor vector" guarantees the performance transfer from pre-training to the downstream task. Our contributions can be summarized as follows:

Realistic model. We build the connection between pre-training and downstream tasks in a more realistic way. We use a log-linear model for the word probabilities in the data, which aligns with the softmax layer at the end of recent large-scale language models. This opens the black box of the pre-trained model at the last layer. Besides, we focus on the representations learned by the pre-trained models, which contain more information than the probability outputs and could reflect more fundamental properties of the data. We also incorporate the change of input distribution from pre-training to downstream tasks, including the changes caused by prompt engineering, in our results.

Necessary conditions for performance guarantee. We constructed two counterexamples to showcase when the logits learned by our student model can perform badly in the downstream tasks. These counterexamples indicate two necessary conditions for the ground-truth function f^* to be provably learned: (i) f^* should not distinguish between words with small probabilities and words with super-small probabilities (ii) the logits must have some structures to deal with the shift-invariance of the softmax function. Without any of these two properties, it is possible that the representations from the pre-trained model are useless in the downstream task.

Novel observation about representations. To understand the empirical success of performance transfer from pre-training to the downstream task, we hypothesize that there exists an "anchor vector" in the representation space that can be used to es-

timate the bulk partition function, which is the sum of the exponential of all logits except the largest few, i.e., $\sum_{j:z_j^* \text{ not large}} e^{z_j^*}$. We also verified the existence of this vector in various large-scale pre-trained language models.

Sufficient condition and explanation of empirical success. We theoretically showed that the existence of an "anchor vector" is a sufficient condition for a sparse one-hidden-layer ReLU network f^* to be learned by our student model f. Specifically, assuming that f^* is a one-hidden-layer ReLU network depending on a small set of words and the downstream task is a binary classification depending on $f^*(z^*)$, the existence of the anchor vector enable us to upper bound the downstream task performance of the student model $f(z)$ by its KL divergence in pre-training. In other words, a small pre-training loss is guaranteed to transfer to a small downstream classification error.

This paper is organized as follows: We discuss the related works in Section 3.1.2. In Section 3.2, we provide the detail about the problem setting. We then construct counterexamples to present the two necessary conditions for the performance transfer in Section 3.3. The hypothesis about the representation structure and the empirical verifications are in Section 3.4 and the formal proof of sufficiency follows in Section 3.5. We finally conclude our paper and discuss future directions in Section 3.6.

3.1.2 Related Works

Theoretical understanding why pre-training helps downstream tasks. Most of the works along this line rely on latent variable models and show that pre-training could recover some form of the latent variables. Arora et al. (2016) proposed the RAND-WALK latent model and explain the empirical success of word embedding methods such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). Arora et al. (2017) extended the previous model to justify sentence embeddings theoretically, and

Arora et al. (2018a) explained sentence embedding via compressed sensing. Other models are also used in this line of work, e.g., Hidden Markov Model (Wei et al., 2021) and graphical model (Zhang and Hashimoto, 2021). Besides, (Lee et al., 2021; Tosh et al., 2021b) assume the conditional independence or multi-view structure in the pre-training data and prove that training an additional linear layer on top of learned representations could perform well in downstream tasks.

The problem setting in our paper is similar to that of Nikunj et al. (2021), which also analyzes the performance transfer of pre-trained language models to binary classification downstream tasks. They treat the pre-trained model as a black box and assume that the downstream task can be formulated into a sentence completion task, while we open the black box at the last layer and connect pre-training with downstream tasks by the "anchor vector" and function f^*. Besides, they focus on the prediction probabilities of the pre-trained model while we focus on the representations instead.

Applications and analysis of hidden representations from large-scale language models. The hidden representations produced by large language models like BERT (Devlin et al., 2018) or ELMo (Peters et al., 2018) have been very useful in various NLP tasks. A standard way is to train a linear classifier on these representations, and there are other ways such as using the normalized mean of concatenated word embeddings (Tanaka et al., 2020). To understand why these word embeddings are useful, people have empirically showed that BERT word embeddings contain information about sentence-level context (Miaschi and Dell'Orletta, 2020), word sense (Wiedemann et al., 2019), syntactic phenomena (Tenney et al., 2019) including parse trees (Hewitt and Manning, 2019; Kim et al., 2020).

Language modeling can also be considered as a form of applying the hidden representations because the next word probability is usually the product of the representation and the dictionary matrix going through softmax. Therefore, any zero-shot application of pre-trained auto-regressive language models, e.g., GPT-3 (Brown et al., 2020b) and T5 (Raffel et al., 2020b), is a specific form of using the hidden representations.

3.2 Problem Setup

Notations. We use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. For an input sequence $x = (x_1, \ldots, x_L)$, we use x_{-i} to denote the input sequence without the i-th entry where $i \in [L]$, i.e., $x_{-i} := (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_L)$. We let $D_{KL}(p||q)$ be the KL-divergence between distributions p and q and define $H(p)$ to be the entropy of distribution p.

Data Generation. We consider the following model: There is a set of words $[n]$, each with a fixed corresponding vector $v_j^* \in \mathbb{R}^d$ ($j \in [n]$). We refer to each v_j^* as an atom and the matrix $[v_1^*, \cdots, v_n^*] \in \mathbb{R}^{d \times n}$ as the dictionary. At each position i, let x_i be the value of the word at that position, then $x_i \in [n]$. Assume that the probability of x_i given x_{-i} follows a log-linear model, i.e.,

$$p^*(x_i = j | x_{-i}) \propto \exp\big(\langle v_{-i}^*(x_{-i}), v_j^* \rangle\big), \tag{3.1}$$

where $v_{-i}^*(\cdot)$ is a function that encodes the remaining sequence x_{-i} into a vector in $\mathbb{R}^d$.

We also use $z_j^*(x, i) := \langle v_{-i}^*(x_{-i}), v_j^* \rangle$ to denote the j-th logit and $Z^*(x, i) := \sum_{j=1}^n \exp\big(z_j^*(x, i)\big)$ to denote the partition function, i.e., the normalization factor of

equation (3.1). In other words,

$$\forall j \in [n], \quad p^*(x_i = j | x_{-i}) = \frac{\exp\big(z_j^*(x,i)\big)}{Z^*(x,i)} = \frac{\exp\big(\langle v_{-i}^*(x_{-i}), v_j^* \rangle\big)}{Z^*(x,i)}. \tag{3.2}$$

Student Model. We use a black-box neural network model whose penultimate layer outputs a d'-dimensional vector $v_{-i}(x_{-i}) \in \mathbb{R}^d$, and the last layer is a fully-connected layer with weight matrix $[v_1, v_2, \cdots, v_n] \in \mathbb{R}^{d' \times n}$ followed by the softmax function. In other words, the model output is

$$p(x_i = j | x_{-i}) \propto \exp(\langle v_{-i}(x_{-i}), v_j \rangle). \tag{3.3}$$

Similar to the previous section, we use $z_j(x,i) := \langle v_{-i}(x_{-i}), v_j \rangle$ to denote the j-th logit and $Z(x,i) := \sum_{j=1}^n \exp(z_j(x,i))$ to denote the partition function of equation (3.3), so

$$\forall j \in [n], \quad p(x_i = j | x_{-i}) = \frac{\exp(\langle v_{-i}(x_{-i}), v_j \rangle)}{Z(x,i)}. \tag{3.4}$$

Pre-training. For self-supervised pre-training, we are given (data, "label") pairs (x_{-i}, x_i), and want our model to predict the "label" x_i given x_{-i}. The pre-training loss we use here is cross-entropy loss:

$$\begin{aligned} \ell(v_{-i}) &= \mathbb{E}_x[-p^*(x_i|x_{-i}) \log p(x_i|x_{-i})] \\ &= \mathbb{E}_x[D_{KL}(p^*(x_i|x_{-i}) || p(x_i|x_{-i}))] + \mathbb{E}_x[H(p^*(x_i|x_{-i}))]. \end{aligned}$$

Note that $\mathbb{E}_x[H(p^*(x_i|x_{-i}))]$ is a constant, and we assume that our student model achieves a small loss so that the KL-divergence term $\mathbb{E}_x[D_{KL}(p^*(x_i|x_{-i}) || p(x_i|x_{-i}))] \leqslant \epsilon_{\mathrm{KL}}$ for some ϵ_{KL}

Downstream Task. The downstream task we are considering is binary sequence classification, e.g., sentiment classification. For instance, give a sentence/paragraph/article, we (perhaps after adding some prompts) use our pre-trained model to predict the missing word x_i from the given input x_{-i}. We assume that there is a perfect classifier $f^*(x, i)$ only depending on v^*_{-i} that can distinguish between positive and negative samples. In other words, $\forall x \in \text{POS}, f^*(x, i) > 0$ and $\forall x \in \text{NEG}, f^*(x, i) < 0$. Here POS and NEG are the set of positive and negative input sequences, separately.

A simple downstream task is one whose classifier is linear in v^*_{-i}, that is, $f^*(x, i) := \langle v^*_{-i}(x_{-i}), u^* \rangle$ for some $u^* \in \mathbb{R}^d$. One might also expect more structures in the vectors u^* and v^*_{-i}. For u^*, the downstream task usually depends on only a small set of the words which are related to this task. For example, for sentiment classification, the sentiment of a sentence depends mostly on words similar to "positive" or "negative". This can be formalized as the following assumption:

Assumption 3.2.1 (u^* is a k-sparse combination of $\{v^*_j\}_{j=1}^n$)**.** Assume u^* a sparse combination of at most k vectors in $\{v^*_j\}_{j=1}^n$ and WLOG assume these k vectors are $\{v_1, \ldots, v_k\}$, i.e., there exist coefficients $\{c_j\}_{j=1}^k \in \mathbb{R}^k$ such that $u^* = \sum_{t=1}^k c^*_t v^*_t$.

As for v^*_{-i}, we could add a prompt to the input sequence, which has become the dominant way of using large language models for downstream tasks (Brown et al., 2020b; Radford et al., 2019). For instance, for movie review classification, appending "This movie was" to the original input could improve the classification accuracy. Adding prompts to the input would make the distribution of inputs in the downstream task different from that in pre-training. The difference in distribution can also come from the source of data. For example, the samples in the downstream task may only contain movie reviews while the pre-training dataset can include all kinds of texts on the Internet. Similar to the assumption made in Nikunj et al. (2021),

we use $\mu \in (0, 1]$ to capture this difference. A smaller μ indicates a larger difference between the two distributions, and $\mu = 1$ if and only if these two distributions are the same.

Assumption 3.2.2 (Difference between pre-training and downstream distribution)**.** Let p_{pre} and p_{DS} be the probability density functions of the pre-training and downstream task, respectively. We assume that there exists $\mu \in (0, 1]$ such that

$$\forall i, \forall x \in \text{POS} \cup \text{NEG},\ p_{pre}(x_{-i}) \geqslant \mu \cdot p_{DS}(x_{-i}).$$

Ideally, we want to show that such simple downstream tasks can also be solved well with the representations learned by our student model, i.e., v_{-i} and $\{v_j\}_{j=1}^{k}$. However, in Section 3.3 we show that this is not the case. More structure in the model and downstream task is necessary to make sure that the pre-trained representations are useful for the downstream task.

3.3 Cases When Representations are Insufficient in Downstream Tasks

The problem setting in Section 3.2 seems reasonable at first sight, but in the following subsections, we will show that this model is not enough to guarantee good performance in pre-training generalizes to downstream tasks. In other words, there are ways for the student model to approximate the ground-truth probabilities very well in terms of KL divergence but perform very badly at the downstream task. Therefore, we need to put further constraints on the ground-truth model and the downstream task.

3.3.1 Downstream Tasks Sensitive to Words with Super-Small Probabilities

Intuitively, KL divergence is a weighted log probability difference between two distributions where the weight is the ground-truth probability. Therefore, for the entries with small ground-truth probabilities, a large log probability difference will not result in a large KL divergence. However, the log probability difference is proportional to the difference in the value of $f^*(x, i)$. This makes it possible for the student model to flip the sign of $f^*(x, i)$ without incurring a large KL divergence, as presented in Theorem 3.3.1 whose proof is given in Appendix B.1.

Theorem 3.3.1. *Suppose the downstream task performance depends only on a function* $f^*(x, i) = \langle v^*_{-i}(x_{-i}), u^* \rangle = \sum_{t=1}^{k} c^*_t \langle v^*_{-i}(x_{-i}), v^*_t \rangle$. *For* $t^- \in [k]$, *define* $p^- := p^*(x_i = t^- | x_{-i})$, *and assume* $p^- \leqslant \frac{1}{2}$. *Then for all* $s \in \mathbb{R}^+$, *there exist functions* v_{-i} *and* $\{v_t\}_{t=1}^k$ *such that* $D_{\mathrm{KL}}(p^*(x_i|x_{-i})||p(x_i|x_{-i})) \leqslant 2sp^-$ *and* $f(x, i) := \sum_{t=1}^{k} c^*_t \langle v_{-i}(x_{-i}), v_t \rangle \leqslant f^*(x, i) - s \cdot c^*_{t^-}$.

Theorem 3.3.1 shows that if there is some word of interest t^- that has a small probability p^-, then it is possible to have a model with small KL divergence in pre-training but bad downstream performance. This is because changing the KL divergence by only $2p^- \cdot \frac{f^*(x,i)}{c^*_{t^-}}$ is enough to change the label of the downstream prediction. In other words, as long as the KL divergence is higher than the threshold $2p^- \cdot \frac{f^*(x,i)}{c^*_{t^-}}$, we cannot distinguish between the case when the student model makes an already small probability even smaller (which will probably hurt the downstream task performance) and the case when we are just having some random approximation errors across the entries. In this case, a small KL divergence does not necessarily imply good downstream performance.

Note that this sensitivity of the downstream task to the logits of very small

probability words is not natural. For the downstream tasks in practice, whether a word has a probability of 10^{-5} or 10^{-10} should not influence the label of the sequence. Thus, we need to impose addition structure of our model. We make the downstream task ignore super-small entries by setting a threshold for the logits and ignore the logits smaller than that threshold. In this case, making the logits smaller when it's already small will have no influence on the downstream task performance. Concretely, the enhanced model will be

$$f^*(x,i) = \sum_{j=1}^{k} a_j^* \sigma(z_j^*(x,i) - b_j^*) = \sum_{j=1}^{k} a_j^* \sigma(\langle v_{-i}^*(x_{-i}), v_j^* \rangle - b_j^*). \tag{3.5}$$

3.3.2 Representations are not Shift-Invariant

The softmax function is invariant under shift, i.e., the output stays the same if we add the same value to every coordinate of the input. In the current model, We have no control over the shift of student model logits on unseen data. Consequently, even if we get a student model that performs well on the training data for the downstream task, we cannot guarantee the performance of this model on new data. This can be formalized in the following theorem.

Theorem 3.3.2. *Assume the logits $z^*(x,i)$ are bounded. For any function $f^*(x,i) = \sum_{j=1}^{n} a_j \sigma(z_j^*(x,i) - b_j)$, there exist functions $\{\widehat{z}_j(x,i)\}_{j=1}^n$ such that for all x and i, we have $\widehat{p}(x_i|x_{-i}) = p^*(x_i|x_{-i})$ and $\widehat{f}(x,i) := \sum_{j=1}^{n} a_j^* \sigma(\widehat{z}_j(x,i) - b_j^*)$ is always equal to 0. In other words, the pre-training loss of the model $\{\widehat{z}_j(x,i)\}_{j=1}^n$ is the same as $\{z_j(x,i)\}_{j=1}^n$, but its logits are useless for the downstream task.*

Proof. We choose $\tau \in \mathbb{R}$ such that $\forall x, i, \tau < \min_{j\in[n]} b_j^* - \max_{j\in[n]} z_j^*(x,i)$, and

$\forall x, i, \forall j \in [n]$, we set $\widehat{z}_j(x,i) := z_j^*(x,i) + \tau$, then

$$\forall j \in [n], \widehat{z}_j(x,i) - b_j^* < z_j^*(x,i) + \min_{j\in[n]} b_j^* - \max_{j\in[n]} z_j^*(x,i) - b_j^* \leqslant 0, \tag{3.6}$$

which implies that $\sigma(\widehat{z}_j(x,i) - b_j^*) = 0$. Therefore, $\forall x, i$, we have $\widehat{f}(x,i) = 0$. □

Theorem 3.3.2 indicates that without any structure in the representations, the student model is able to shift the logits for any sample and keep the pre-training loss unchanged. In the worst case, it can shift the logits for unseen data drastically, resulting in a bad downstream performance. Therefore, a theoretical guarantee in downstream performance requires structures in the representations learned by the pre-trained model.

3.4 "Anchor Vector" Hypothesis and Empirical Verifications

In Section 3.3.2, we have shown that the shift-invariance of the softmax function could potentially make the student logits useless for the downstream task. Therefore, to understand why the downstream tasks benefit from representations from the pre-trained model, we need to understand the structure of these representations, and this structure should be able to handle the shift-invariance problem.

3.4.1 "Anchor Vector" Hypothesis

There are different ways to prevent the shift-invariance of softmax from influencing the performance of the downstream tasks. One way of doing this is to keep the partition function stable. As shown in (3.4), the probability of a word is the exponential of the corresponding logit divided by the partition function. If the partition function is constant for different samples, the logits can be uniquely determined by the probabilities, which solves the shift-invariance problem. Arora et al. (2016) showed

that when both the word embeddings and the latent representation are uniformly distributed on a sphere, the partition function is close to a constant with high probability. They also empirically verified this uniformity of word embeddings trained using GloVe (Pennington et al., 2014) and word2vec (Mikolov et al., 2013) on English Wikipedia. However, this is not true for recent large-scale pre-trained language models. Empirical evidence about this will be shown in Section 3.4.3.

Instead of uniformity of word embeddings, in large pre-trained models such as GPT-2 (Radford et al., 2019), we observe that if we remove several top logits from the log partition function, the remaining part can be well approximated by the inner product between the hidden representation $v_{-i}(x)$ and a fixed vector. This motivates us to have the following assumption and hypothesis:

Definition 3.4.1. For a sample x and position i, we could select a set of bulk words $B(x,i) \subset [n]$, and we define the bulk partition functions as $Z^*_{\text{bulk}}(x,i) := \sum_{j\in B(x,i)} \exp\big(\langle v^*_{-i}(x_{-i}), v^*_j\rangle\big)$ and $Z_{\text{bulk}}(x,i) := \sum_{j\in B(x,i)} \exp(\langle v_{-i}(x_{-i}), v_j\rangle)$.

The selection of bulk words $B(x,i)$ can usually be selected by a simple algorithm or manually. For instance, we can construct $B(x,i)$ by taking out the words corresponding to the largest entries in $p(x_i|x_{-i})$. We can also manually select all the words that are irrelevant to the downstream task.

Hypothesis 3.4.2 ("Anchor vector" hypothesis)**.** *There exists $v_0 \in \mathbb{R}^d$ such that*

$$\langle v_{-i}(x_{-i}), v_0\rangle \approx \log Z_{\text{bulk}}(x,i).$$

If our hypothesis holds, we can use v_0 as an anchor for the logits because it could be used to estimate the partition function, and this does not change the form of the downstream task by much. In the following subsections, we will show that

this Hypothesis 3.4.2 holds for most popular language models and provide some discussions. More details about model information and performance are provided in Appendix B.3.

3.4.2 Empirical Verification of "Anchor Vector" Hypothesis

Figure 3.1 plots the mean squared approximation error of the log bulk partition function. We use different versions of pre-trained GPT-2 (Radford et al., 2019) and OPT (Zhang et al., 2022b), and use the first 1/4 of WikiText-2 (Merity et al., 2016) as the input text. The hidden representations we use in this experiment are the last hidden states of these models, i.e., the output of the penultimate layer. The dimension of the hidden representations ranges from 768 to 2048, and the number of tokens is about 70k. We choose the bulk words to be all the words except those having top-k probabilities and compute the optimal anchor vector using the closed-form least-squares solution. In our experiments, we use the mean squared error (MSE) to measure the approximation quality. Formally, the MSE is defined as

$$\epsilon_{\text{MSE}} = \min_{v_0} \mathbb{E}_{x,i}[(v_{-i}(x_{-i}), v_0\rangle - \log Z_{\text{bulk}}(x, i))^2].$$

The values of the bulk partition functions are always around 10, and we can see from Figure 3.1 that the MSE is usually several orders of magnitude smaller. Therefore, the inner product between the hidden representation and the optimal anchor vector can usually approximate the log bulk partition function well, and the approximation improves as k increases, i.e., when we ignore more top words. This validates our "anchor vector" hypothesis.

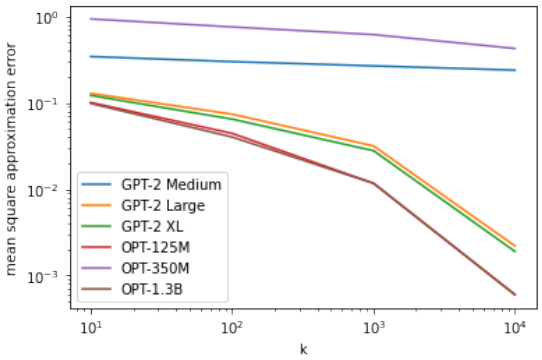


FIGURE 3.1: Mean squared error of log bulk partition function with different k in different models. Bulk partition function includes all logits except top-k ones.

3.4.3 Existence of Anchor Vector is not Trivial

In this section, we will show why it is important to consider the bulk partition function. As discussed in Section 3.4.1, the partition function becomes easy to predict under some settings, e.g., when the word embeddings are uniformly distributed on a sphere. In this section, we will empirically show that the large-scale language models do not fall under these settings. As a result, the existence of this vector is not trivial.

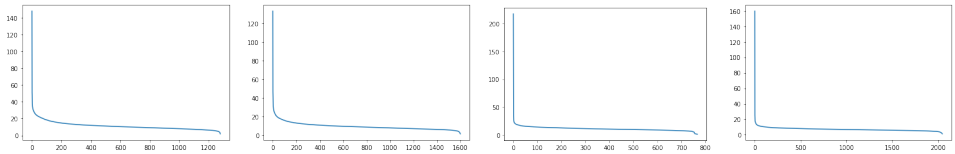
Linear approximation is not accurate for original partition function. Table 3.1 shows the comparison between the MSE of approximating the original partition function and the bulk partition function. We select the four models with the smallest bulk MSE. For these models, although the log bulk partition function can be well approximated linearly, the logarithm of the original partition function cannot. This confirms the necessity of removing the words of interest in the partition function.

Word embeddings are not uniformly distributed on a sphere. Figure 3.2 shows the singular values of dictionary matrices from different models. For the four selected models, their word embeddings are close to being low-rank. Therefore, the word embeddings

Table 3.1: Comparison between MSE of log bulk partition function (except top-100) and log partition function

Model	original MSE	bulk MSE
GPT-2 Large	0.7296	0.0740
GPT-2 XL	0.7716	0.0648
OPT-125M	0.6688	0.0443
OPT-1.3B	0.5234	0.0402

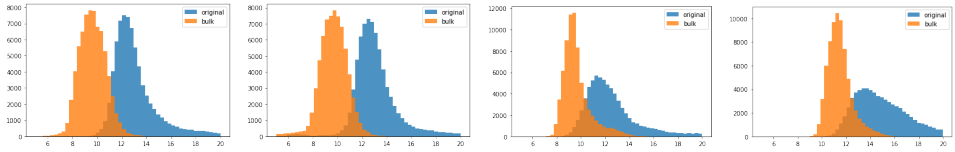
in these models are far from uniformly distributed on a sphere.



(a) GPT-2 Large (b) GPT-2 XL (c) OPT-125M (d) OPT-1.3B

FIGURE 3.2: Singular values of dictionary matrices.

(Bulk) Partition functions have large variations. Figure 3.3 shows the histogram of log partition function and log bulk partition function for the language models, where the bulk words are defined as all the words except the top 100, in terms of logit values. We can see from the figures that both the partition function and the bulk partition function vary a lot depending on the samples. Therefore, in recent large-scale language models, the partition functions are not stable across samples.



(a) GPT-2 Large (b) GPT-2 XL (c) OPT-125M (d) OPT-1.3B

FIGURE 3.3: Histogram of log original and bulk partition functions

3.5 Anchor Vector Guarantees Performance Transfer from Pre-Training to Downstream Tasks

Based on the counterexamples in Section 3.3 and the observations in Section 3.4.1, we now give sufficient conditions on the downstream tasks so that classification accuracy benefits from a pre-trained representation.

3.5.1 Modifications to the Model

Given the anchor vector hypothesis we proposed in Section 3.4: For huge language models, there always exists a vector v_0^* such that its inner product with the hidden state $v_{-i}^*(x_{-i})$ approximates the logarithm of the bulk partition function. Therefore, we could use v_0^* as an anchor to handle the shift invariance problem of the softmax function, i.e., we want to subtract $\langle v_{-i}^*(x_{-i}), v_0^* \rangle$ from $\langle v_{-i}^*(x_{-i}), v_j^* \rangle$. As a result, we modify our model for the downstream task to be

$$f^*(x, i) := \sum_{j=1}^{k} a_j^* \sigma(\langle v_{-i}^*(x_{-i}), v_j^* - v_0^* \rangle - b_j^*). \tag{3.7}$$

The student model is also modified accordingly:

$$f(x, i) := \sum_{j=1}^{k} a_j \sigma(\langle v_{-i}(x), v_j - v_0 \rangle - b_j). \tag{3.8}$$

Note that in the above model we have already adapted Assumption 3.2.1 which was originally stated for the linear model. Therefore, we also update the assumption and restate it below:

Assumption 3.5.1 (At most k words of interest)**.** Assume there are at most k vectors in $\{v_j^*\}_{j=1}^n$ whose logits are relevant to the downstream task and WLOG assume

these k vectors are $\{v_1, \ldots, v_k\}$. In other words, we assume there exist coefficients $\{a_j^*\}_{j=1}^k \in \mathbb{R}^k$ such that $f^*(x, i) := \sum_{j=1}^k a_j^* \sigma(\langle v_{-i}^*(x_{-i}), v_j^* - v_0^* \rangle - b_j^*)$.

We are still assuming the teacher-student setting and the log-linear word production model for the word probabilities, as restated below:

$$p^*(x_i = j | x_{-i}) = \frac{\exp\big(\langle v_{-i}^*(x), v_j^* \rangle\big)}{Z^*}, \quad p(x_i = j | x_{-i}) = \frac{\exp(\langle v_{-i}(x), v_j \rangle)}{Z}. \tag{3.9}$$

3.5.2 Assumptions

Under the modified model defined in Section 3.5.1, we will make some additional assumptions. Firstly, we assume that there is a margin between the ground-truth function values for positive and negative samples.

Assumption 3.5.2 (Margin for downstream task)**.** There exists a margin $\gamma \in \mathbb{R}^+$ such that at any position i, if $x^{\text{pos}} \in \text{POS}$, then $f^*(x^{\text{pos}}, i) \geqslant \gamma$, and if $x^{\text{neg}} \in \text{NEG}$, then $f^*(x^{\text{neg}}, i) \leqslant -\gamma$, where POS and NEG are the sets of positive and negative samples, respectively.

As noticed in the experiments, the log bulk partition function (as defined in Definition 3.4.1) can be linearly approximated by the hidden state. Normally, the bulk only contains words that are not related to the downstream task, and every single word usually has a small probability, but the total probability of the bulk words is not negligible, as reflected in the following two assumptions. Note that the set of bulk words can be strictly contained in the complement of the set of words of interest. This is useful especially when we are not certain about which words are important for the downstream task.

Assumption 3.5.3 (Bulk contains no words of interest)**.** For all x and i, $B(x, i) \cap \{1, \ldots, k\} = \varnothing$.

Assumption 3.5.4 (Lower bound of bulk probability)**.** For all x and i,

$$\frac{Z^*_{\text{bulk}}(x,i)}{Z^*(x,i)} \geqslant p_b.$$

The "anchor" vector $v_0^*(i)$ is used to handle the instability of the partition function, and we formalize our anchor vector hypothesis as the following assumption:

Assumption 3.5.5 (Linear approximation of log bulk partition function)**.** $\exists v_0, v_0^* \in \mathbb{R}^d, \varepsilon_b \in \mathbb{R}^+$ s.t.

$$\forall x, \max\{|\langle v^*_{-i}(x), v_0^*\rangle - \log Z^*_{\text{bulk}}(x,i)|, |\langle v_{-i}(x), v_0\rangle - \log Z_{\text{bulk}}(x,i)|\} \leqslant \varepsilon_b.$$

Furthermore, we assume $\varepsilon_b \leqslant \frac{\gamma}{4k \max_{j\in[k]} |a_j^*|}$.

For notational simplicity, we will use all the notations (including f^*, f, v_0^*, and v_0) without i when the selection of i is clear from the context.

3.5.3 Main Theorem and Interpretations

In our model, the ground-truth function f^* contains k terms with coefficients $\{a_j^*\}_{j=1}^k$ and we define the margin γ as the margin for f^*. If we scale $\{a_j^*\}_{j=1}^k$ or increase k by adding duplicated terms to f^*, we can scale γ arbitrarily without changing the pre-training performance or the downstream prediction of the student model. To construct a quantity that better indicates the difficulty of the downstream task, we have the following definition of the normalized margin that is invariant to the scaling of k and $\{a_j^*\}_{j=1}^k$:

Definition 3.5.6. We define the normalized margin as $\Gamma := \frac{\gamma}{k \max_{j\in[k]} |a_j^*|}$.

Then we are ready to state our main result:

Theorem 3.5.7. *Let* $\epsilon_{\mathrm{KL}} := \mathbb{E}_{x\sim\mathcal{D}_{pre}}[D_{\mathrm{KL}}(p^*(x)||p(x))]$ *be the pre-training loss, and we let* $\epsilon_{CLS} := \Pr_{x\sim\mathcal{D}_{DS}}[f(x)\cdot f^*(x) < 0]$ *be the downstream classification error rate, where* $\mathcal{D}_{pre}$ *and* $\mathcal{D}_{DS}$ *are the input distributions of pre-training and downstream data. Under Assumptions 3.2.2-3.5.5, further assume* $\min_{j\in[k]} b_j^* \geqslant \epsilon_b - \log(4k)$ *and* $8\epsilon_b < \Gamma < 6$*, then there exists a set of parameters* $(a_j)_{j=1}^n, (b_j)_{j=1}^n$ *such that*

$$\epsilon_{CLS} \leqslant \epsilon_{\mathrm{KL}} \cdot \frac{288}{\mu \cdot p_b^2 \cdot \Gamma^2}. \tag{3.10}$$

This theorem shows that when "anchor vector" exists, we could upper bound the downstream classification error by the KL divergence of the student model during pre-training. This upper bound becomes smaller when the distribution for input data of the downstream task is close to that of pre-training, the bulk probability is non-negligible, and the normalized margin of the ground-truth classifier is large. Here we discuss these ways to decrease the upper bound and their corresponding intuitions.

Large μ**.** μ is larger when the data distributions of pre-training and the downstream task become closer, which helps with the performance transfer.

Large p_b**.** The bulk probability $\frac{Z^*_{\mathrm{bulk}}}{Z^*}$ is usually at least a constant in practice. When the bulk probability becomes larger, the anchor vector plays a more important role in the partition function and the downstream task.

Large Γ**.** A larger normalized margin makes it harder for the student model to make mistakes in downstream prediction.

Remark 3.5.8. For ease of presentation, we are making additional assumptions in Theorem 3.5.7, e.g., $\min_{j\in[k]} b_j^* \geqslant \epsilon_b - \log(4k)$ and $4\epsilon_b < \Gamma < 3$. More general cases are covered in Lemma 3.5.9. We also discuss potential ways to improve the bound

in Section 3.5.4.

3.5.4 Proof and Discussions for Main Theorem

Here we provide a proof sketch of Theorem 3.5.7.

Theorem 3.5.7. *Let* $\epsilon_{\mathrm{KL}} := \mathbb{E}_{x\sim\mathcal{D}_{pre}}[D_{\mathrm{KL}}(p^*(x)||p(x))]$ *be the pre-training loss, and we let* $\epsilon_{CLS} := \Pr_{x\sim\mathcal{D}_{DS}}[f(x)\cdot f^*(x) < 0]$ *be the downstream classification error rate, where* $\mathcal{D}_{pre}$ *and* $\mathcal{D}_{DS}$ *are the input distributions of pre-training and downstream data. Under Assumptions 3.2.2-3.5.5, further assume* $\min_{j\in[k]} b_j^* \geqslant \epsilon_b - \log(4k)$ *and* $8\epsilon_b < \Gamma < 6$*, then there exists a set of parameters* $(a_j)_{j=1}^n, (b_j)_{j=1}^n$ *such that*

$$\epsilon_{CLS} \leqslant \epsilon_{\mathrm{KL}} \cdot \frac{288}{\mu \cdot p_b^2 \cdot \Gamma^2}. \tag{3.10}$$

Proof. We have the following lemma providing a lower bound of the TV distance between $p^*(x_i|x_{-i})$ and $p(x_i|x_{-i})$, which will be proved in Section 3.5.5.

Lemma 3.5.9. *There exists a choice of* $(a_j)_{j=1}^k, (b_j)_{j=1}^k$ *such that if Assumptions 3.2.2-3.5.5 hold and there exists* $x \in \mathrm{POS}\cup\mathrm{NEG}$ *and* i *such that* $f(x,i)\cdot f^*(x,i) < 0$*, then we must have*

$$TV(p^*(x_i|x_{-i}), p(x_i|x_{-i})) \tag{3.11}$$

$$\geqslant p_b \min_{\varepsilon_p\geqslant 0}\left\{(1-e^{-\varepsilon_p}) + k\cdot\min_{j\in[k]} e^{b_j^*-\varepsilon_b-\varepsilon_p}\cdot\left(e^{\sigma\left(\frac{\Gamma}{2}-2\varepsilon_b-\varepsilon_p\right)}-1\right)\right\}. \tag{3.12}$$

From Lemma 3.5.9 and Pinsker's inequality we can lower bound the KL divergence

at any incorrectly classified sample: For all $x \in \text{POS} \cup \text{NEG}$,

$$\text{KL}(p^*(x_i|x_{-i})||p(x_i|x_{-i})) \tag{3.13}$$

$$\geqslant 2(\text{TV}(p^*(x_i|x_{-i}), p(x_i|x_{-i})))^2 \tag{3.14}$$

$$\geqslant 2p_b^2 \min_{\varepsilon_p \geqslant 0} \left\{ (1 - e^{-\varepsilon_p}) + k \cdot \min_{j \in [k]} e^{b_j^* - \varepsilon_b - \varepsilon_p} \cdot \left(e^{\sigma\left(\frac{\Gamma}{2} - 2\varepsilon_b - \varepsilon_p\right)} - 1 \right) \right\}^2 \tag{3.15}$$

Since the pre-training loss is the expected KL divergence for pre-training data, it is lower bounded by the KL divergence on the incorrectly classified samples, which is related to the downstream classification error rate:

$$\epsilon_{\text{KL}} = \mathbb{E}_{x \sim \mathcal{D}_{pre}}[\text{KL}(p^*(x_i|x_{-i})||p(x_i|x_{-i}))] \tag{3.16}$$

$$\geqslant \Pr_{x_{-i} \sim \mathcal{D}_{pre}}[f(x,i) \cdot f^*(x,i) < 0] \cdot \min_{x_{-i}: f(x,i) \cdot f^*(x,i) < 0} \text{KL}(p^*(x_i|x_{-i})||p(x_i|x_{-i})) \tag{3.17}$$

$$\geqslant \mu \cdot \Pr_{x_{-i} \sim \mathcal{D}_{DS}}[f(x,i) \cdot f^*(x,i) < 0] \cdot \min_{x_{-i}: f(x,i) \cdot f^*(x,i) < 0} \text{KL}(p^*(x_i|x_{-i})||p(x_i|x_{-i})) \tag{3.18}$$

$$\geqslant \epsilon_{CLS} \cdot 2\mu \cdot p_b^2 \min_{\varepsilon_p \geqslant 0} \left\{ (1 - e^{-\varepsilon_p}) + k \cdot \min_{j \in A(i)} e^{b_j^* - \varepsilon_b - \varepsilon_p} \cdot \left(e^{\sigma\left(\frac{\Gamma}{2} - 2\varepsilon_b - \varepsilon_p\right)} - 1 \right) \right\}^2. \tag{3.19}$$

Here $\mathcal{D}_{pre}$ and $\mathcal{D}_{DS}$ are the data distributions of pre-training and downstream task, and the second inequality comes from Assumption 3.2.2.

Now we have established the relationship between ϵ_{KL} and ϵ_{CLS}, and the only step left is to bound the minimum over ε_p. For notation simplicity, we define $\Lambda := \frac{\Gamma}{2} - 2\epsilon_b$.

If $\varepsilon_p \leqslant \frac{\Lambda}{3}$, we have $(1 - e^{-\varepsilon_p}) \geqslant (1 - e^{-\frac{\Lambda}{3}})$.

If $\varepsilon_p > \frac{\Lambda}{3}$, we know that

$$e^{b_j^* - \varepsilon_b - \varepsilon_p} \cdot \left(e^{\frac{\gamma}{2k \max_{j \in A(i)} |a_j^*|} - 2\varepsilon_b - \varepsilon_p} - 1 \right) > e^{b_j^* - \varepsilon_b - \frac{\Lambda}{3}} \cdot \left(e^{\frac{2\Lambda}{3}} - 1 \right) = e^{b_j^* - \varepsilon_b} \cdot \sinh\left(\frac{\Lambda}{3}\right).$$

Thus,

$$\min_{\varepsilon_p \geqslant 0} \left\{ (1 - e^{-\varepsilon_p}) + k \cdot \min_{j \in A(i)} e^{b_j^* - \varepsilon_b - \varepsilon_p} \cdot \left(e^{\sigma\left(\frac{\Gamma}{2} - 2\varepsilon_b - \varepsilon_p\right)} - 1 \right) \right\} \tag{3.20}$$

$$\geqslant \min \left\{ 1 - e^{-\frac{\Lambda}{3}}, 2k \cdot \min_{j \in A(i)} e^{b_j^* - \varepsilon_b} \cdot \sinh\left(\frac{\Lambda}{3}\right) \right\} \tag{3.21}$$

Plugging this into (3.19) gives us

$$\epsilon_{\mathrm{KL}} \geqslant \epsilon_{CLS} \cdot 2\mu \cdot p_b^2 \min \left\{ (1 - e^{-\frac{\Lambda}{3}})^2, 4k^2 \cdot \min_{j \in [k]} e^{2b_j^* - 2\varepsilon_b} \cdot \sinh^2(\frac{\Lambda}{3}) \right\}. \tag{3.22}$$

Since $\Gamma > 8\epsilon_b$, we know that $\frac{\Gamma}{4} < \Lambda < 3$, so $(1 - e^{-\frac{\Lambda}{3}})^2 > \left(\frac{\Lambda}{6}\right)^2 > \frac{\Gamma^2}{576}$.

Besides, $\min_{j \in [k]} b_j^* \geqslant \epsilon_b - \log(4k)$ implies $4k^2 \cdot \min_{j \in [k]} e^{2b_j^* - 2\varepsilon_b} \geqslant \frac{1}{4}$. We also have $\sinh^2(\frac{\Lambda}{3}) > \left(\frac{\Lambda}{3}\right)^2 > \frac{\Gamma^2}{144}$.

Thus, $\min \left\{ (1 - e^{-\frac{\Lambda}{3}})^2, 4k^2 \cdot \min_{j \in [k]} e^{2b_j^* - 2\varepsilon_b} \cdot \sinh^2(\frac{\Lambda}{3}) \right\} > \frac{\Gamma^2}{576}$. Plugging this into (3.22) finishes the proof of Theorem 3.5.7.

□

Further Discussion of the Bound

Potentially improving the upper bound by better thresholding. We are taking the minimum over two terms in the proof of Theorem 3.5.7. These two terms come from two parts of the difference between the ground-truth probability and our learned probability: the bulk part and the top part. Both parts are functions of ϵ_p, which measures the approximation error for the bulk probability. A large ϵ_p will incur a large error in the bulk part while a smaller ϵ_p will make the top part larger. We are setting $\frac{\Lambda}{3}$ as the threshold for the current bound in Theorem 3.5.7. The bound could be improved by choosing other thresholds if we know more about the values of $\{b_j^*\}_{j=1}^k$ and the normalized margin $\frac{\gamma}{2k \max_{j \in A(i)} |a_j^*|}$.

More general cases for thresholds $\{b_j^*\}_{j=1}^k$ and normalized margin Γ. The thresholds $\{b_j^*\}_{j=1}^k$ can be considered as the model's sensitivity to the logits. A smaller threshold indicates a higher sensitivity. In the proof of Theorem 3.5.7, we are assuming that the thresholds are not very small. In general cases where the thresholds can be small, an important observation is that: For the same set of samples, decreasing the thresholds by some number s will increase the normalized margin by s as well. Since both the thresholds and the normalized margin are in the exponent in our bound (sinh function is close to exponential when Γ is large), these two effects can cancel out to a large degree and don't influence the bound by much.

Prompt engineering can help with downstream performance. Prompt engineering can help with the downstream task performance in multiple ways. The direct way is to make the data distribution closer to that of the pre-training stage to increase μ. Furthermore, it can also indirectly help improve the bound by decreasing the number of activated neurons, increasing the margin, etc.

3.5.5 Proof of Lemma 3.5.9

Here we provide a proof of Lemma 3.5.9, with the proofs of the necessary technical lemmas postponed to Appendix B.2.

Proof. For all $j \in [n]$, we set $a_j = a_j^*$ and $b_j = b_j^*$. Since i is fixed in the proof, we will omit i when the selection of i is clear from context.

WLOG, assume $x \in \text{POS}$ because otherwise we can flip the sign of all a_j^*'s. In this case, from Assumption 3.5.2 we know that $f^*(x,i) \geqslant \gamma$. Therefore, $f(x,i) \cdot f^*(x,i) < 0$ implies $f(x,i) < 0$, so $f^*(x,i) - f(x,i) > \gamma$.

There are two possible ways to make $f(x,i)$ smaller than $f^*(x,i)$: Making the

neurons with positive coefficients smaller or making those with negative coefficients larger. The total difference must be at least γ, so there must exist one sign whose change in value is at least $\frac{\gamma}{2}$. Formally, we decompose the difference $f^*(x,i) - f(x,i)$ into two terms:

$$f^*(x,i) - f(x,i) = \sum_{j:a_j^*>0} a_j^* \left(\sigma(\langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^*) - \sigma(\langle v_{-i}(x), v_j - v_0\rangle - b_j^*)\right) \tag{3.23}$$

$$+ \sum_{j:a_j^*<0} |a_j^*| \left(\sigma(\langle v_{-i}(x), v_j - v_0\rangle - b_j^*) - \sigma(\langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^*)\right). \tag{3.24}$$

Note that at least one of the two terms on the right-hand side must be at least $\frac{\gamma}{2}$ because their sum is at least γ. We first consider the first case, in other words, $\sum_{j:a_j^*>0} a_j^* \left(\sigma(\langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^*) - \sigma(\langle v_{-i}(x), v_j - v_0\rangle - b_j^*)\right) \geqslant \frac{\gamma}{2}$. The analysis of the second term is almost the same as the first one.

For notational simplicity, we define

$$\Delta_j := (\langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^*) - \sigma(\langle v_{-i}(x), v_j - v_0\rangle - b_j^*), \tag{3.25}$$

(n.b. σ only appears in the second term) which implicitly depends on x, and and define the set

$$S(x) := \{j : a_j^* > 0, \Delta_j > 0\}. \tag{3.26}$$

If $\Delta_j > 0$, we must have $\langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^* > 0$ because ReLU is non-negative. This means that the neurons corresponding to the words in $S(x)$ must be activated.

Intuitively, $S(x)$ is the set of words whose corresponding neuron in our student model has a smaller value than the ground-truth model, and Δ_j for a word $j \in S(x)$ is the difference between these two neurons. In other words, if we want to make a

mistake in classifying a sample x, Δ_j is the obstacle for the neuron to overcome, and the sum of all these obstacles must be at least the margin γ. Formally, we have the following lemma for $S(x)$ and lower bound for Δ_j:

Lemma 3.5.10. *$S(x) \neq \varnothing$.*

Lemma 3.5.11. *$\sum_{j\in S(x)} \Delta_j \geqslant \frac{\gamma}{2\max_{j\in S(x)} a_j^*}$.*

Considering the TV distance, we get

$$\sum_{j=1}^{n} |p^*(x_i = j|x_{-i}) - p(x_i = j|x_{-i})| \tag{3.27}$$

$$\geqslant \sum_{j\in B(x,i)} |p^*(x_i = j|x_{-i}) - p(x_i = j|x_{-i})| + \sum_{j\in S(x)} |p^*(x_i = j|x_{-i}) - p(x_i = j|x_{-i})| \tag{3.28}$$

$$\geqslant \left| \sum_{j\in B(x,i)} p^*(x_i = j|x_{-i}) - \sum_{j\in B(x,i)} p(x_i = j|x_{-i}) \right| \tag{3.29}$$

$$+ \sum_{j\in S(x)} |p^*(x_i = j|x_{-i}) - p(x_i = j|x_{-i})| \tag{3.30}$$

$$= \left| \frac{Z^*_{\text{bulk}}(x,i)}{Z^*(x,i)} - \frac{Z_{\text{bulk}}(x,i)}{Z(x,i)} \right| + \sum_{j\in S(x)} |p^*(x_i = j|x_{-i}) - p(x_i = j|x_{-i})|, \tag{3.31}$$

where the first inequality comes from Assumption 3.5.3, i.e., $S(x) \subseteq [k] \subseteq [n]\backslash B(x,i)$.

We define $\varepsilon_p := \left| \log \frac{Z^*_{\text{bulk}}(x,i)}{Z^*(x,i)} - \log \frac{Z_{\text{bulk}}(x,i)}{Z(x,i)} \right|$, which is the approximation error of the log bulk probability and an important quantity to trade off the two terms in the TV distance. There are two terms in (3.31). The first term corresponds to the bulk probability and the second term corresponds to the activated neurons. When ε_p is large, the bulk probability has a large approximation error, resulting in a large TV distance. We lower bound this term in the following lemma:

Lemma 3.5.12.

$$\left|\frac{Z^*_{\text{bulk}}(x,i)}{Z^*(x,i)} - \frac{Z_{\text{bulk}}(x,i)}{Z(x,i)}\right| \geqslant p_b(1-e^{-\varepsilon_p}). \tag{3.32}$$

The right-hand side of the above lemma is an increasing function of ε_p, and this lower bound becomes large when the approximation error of the bulk probability is large.

In the other regime when ε_p is small, the bulk partition function must be approximated accurately, i.e., $\frac{Z^*_{\text{bulk}}(x,i)}{Z^*(x,i)} \approx \frac{Z_{\text{bulk}}(x,i)}{Z(x,i)}$. Thus,

$$|p^*(x_i = j|x_{-i}) - p(x_i = j|x_{-i})| = \left|\frac{\exp\big(\langle v^*_{-i}(x), v^*_j\rangle\big)}{Z^*(x,i)} - \frac{\exp(\langle v_{-i}(x), v_j\rangle)}{Z(x,i)}\right| \tag{3.33}$$

$$\approx \frac{Z^*_{\text{bulk}}(x,i)}{Z^*(x,i)}\left|\frac{\exp\big(\langle v^*_{-i}(x), v^*_j\rangle\big)}{Z^*_{\text{bulk}}(x,i)} - \frac{\exp(\langle v_{-i}(x), v_j\rangle)}{Z_{\text{bulk}}(x,i)}\right|. \tag{3.34}$$

From Assumption 3.5.5 we know the bulk partition functions can be accurately approximated using the vectors v^*_0 and v_0. Thus, $\frac{\exp\big(\langle v^*_{-i}(x), v^*_j\rangle\big)}{Z^*_{\text{bulk}}(x,i)} \approx \exp\big(\langle v^*_{-i}(x), v^*_j - v^*_0\rangle\big)$. So this difference is approximately $|\exp\big(\langle v^*_{-i}(x), v^*_j - v^*_0\rangle\big) - \exp(\langle v_{-i}(x), v_j - v_0\rangle)| \approx \exp(\langle v_{-i}(x), v_j - v_0\rangle)(e^{\Delta_j} - 1)$. As ε_p becomes smaller, the aforementioned approximations become more accurate, making the TV distance suffer from a term that is roughly proportional to $(e^{\Delta_j} - 1)$. This is formalized in the following lemma:

Lemma 3.5.13. *For all $j \in S(x)$,*

$$|p^*(x_i = j|x_{-i}) - p(x_i = j|x_{-i})| \geqslant e^{b^*_j - \varepsilon_b - \varepsilon_p} \cdot p_b \cdot \left(e^{\sigma(\Delta_j - 2\varepsilon_b - \varepsilon_p)} - 1\right). \tag{3.35}$$

Plugging Lemma 3.5.12 and 3.5.13 into (3.31) gives us

$$\mathrm{TV}(p^*(x_i|x_{-i}), p(x_i|x_{-i})) \tag{3.36}$$

$$\geqslant \left| \frac{Z^*_{\mathrm{bulk}}(x,i)}{Z^*(x,i)} - \frac{Z_{\mathrm{bulk}}(x,i)}{Z(x,i)} \right| + \sum_{j \in S(x)} \left| \frac{\exp\big(\langle v^*_{-i}(x), v^*_j \rangle\big)}{Z^*(x,i)} - \frac{\exp(\langle v_{-i}(x), v_j \rangle)}{Z(x,i)} \right| \tag{3.37}$$

$$\geqslant p_b(1 - e^{-\varepsilon_p}) + \sum_{j \in S(x)} e^{b^*_j - \varepsilon_b - \varepsilon_p} \cdot p_b \cdot \left(e^{\sigma(\Delta_j - 2\varepsilon_b - \varepsilon_p)} - 1 \right) \tag{3.38}$$

$$\geqslant p_b(1 - e^{-\varepsilon_p}) + p_b \min_{j \in S(x)} e^{b^*_j - \varepsilon_b - \varepsilon_p} \cdot \sum_{j \in S(x)} \left(e^{\sigma(\Delta_j - 2\varepsilon_b - \varepsilon_p)} - 1 \right) \tag{3.39}$$

$$\geqslant p_b(1 - e^{-\varepsilon_p}) + p_b \min_{j \in S(x)} e^{b^*_j - \varepsilon_b - \varepsilon_p} \cdot |S(x)| \cdot \left(e^{\sum_{j \in S(x)} \sigma(\Delta_j - 2\varepsilon_b - \varepsilon_p)/|S(x)|} - 1 \right), \tag{3.40}$$

where the last inequality comes from Jensen's inequality.

Let $g(u,v) := u \cdot (e^{\frac{v}{u}} - 1)$ for $u, v > 0$, then $\frac{\partial g}{\partial u} = e^{\frac{v}{u}} \left(1 - \frac{v}{u} - e^{-\frac{v}{u}}\right) \leqslant 0$, so from $|S(x)| \leqslant k$ we know that

$$|S(x)| \cdot \left(e^{\sum_{j \in S(x)} \sigma(\Delta_j - 2\varepsilon_b - \varepsilon_p)/|S(x)|} - 1 \right) \geqslant k \cdot \left(e^{\sum_{j \in S(x)} \sigma(\Delta_j - 2\varepsilon_b - \varepsilon_p)/k} - 1 \right).$$

Using the fact that $\sum_i \sigma(y_i) = \sum_i \max\{y_i, 0\} \geqslant \max \boldsymbol{c} \sum_i y_i, 0 = \sigma\left(\sum_i y_i\right)$, we have

$$\sum_{j \in S(x)} \sigma\left(\Delta_j - 2\varepsilon_b - \varepsilon_p\right) \geqslant \sigma\left(\sum_{j \in S(x)} \left(\Delta_j - 2\varepsilon_b - \varepsilon_p\right) \right) \tag{3.41}$$

$$\geqslant \sigma\left(\frac{\gamma}{2 \max_{t \in S(x)} a^*_t} - 2k \cdot \varepsilon_b - k\varepsilon_p \right), \tag{3.42}$$

where the last inequality follows from Lemma 3.5.11. Therefore,

$$\mathrm{TV}(p^*(x_i|x_{-i}), p(x_i|x_{-i})) \tag{3.43}$$

$$\geqslant p_b(1 - e^{-\varepsilon_p}) + p_b \cdot k \min_{j \in S(x)} e^{b^*_j - \varepsilon_b - \varepsilon_p} \cdot \left(e^{\sigma\left(\frac{\gamma}{2k \max_{t \in S(x)} a^*_t} - 2 \cdot \varepsilon_b - \varepsilon_p \right)} - 1 \right). \tag{3.44}$$

Since we do not have any assumption for ε_p, we take the minimum over all possible values of ε_p and get the following bound:

$$\mathrm{TV}(p^*(x_i|x_{-i}), p(x_i|x_{-i})) \tag{3.45}$$

$$\geqslant p_b \min_{\varepsilon_p \geqslant 0} \left\{ (1 - e^{-\varepsilon_p}) + k \cdot \min_{j \in S(x)} e^{b_j^* - \varepsilon_b - \varepsilon_p} \cdot \left(e^{\sigma\left(\frac{\gamma}{2k \max_{t \in S(x)} a_t^*} - 2 \cdot \varepsilon_b - \varepsilon_p \right)} - 1 \right) \right\}. \tag{3.46}$$

Above we have derived a lower bound for the TV distance between p^* and p when the first term of (3.24) is at least $\frac{\gamma}{2}$. When the second term of (3.24) is at least $\frac{\gamma}{2}$, the proof is symmetric to that of the first term and we only need to exchange the role of parameters from the teacher model and student model and the related definitions. Therefore, with the second term being at least $\frac{\gamma}{2}$, we have

$$\mathrm{TV}(p^*(x_i|x_{-i}), p(x_i|x_{-i})) \tag{3.47}$$

$$\geqslant p_b \min_{\varepsilon_p \geqslant 0} \left\{ (1 - e^{-\varepsilon_p}) + k \cdot \min_{j \in S'(x)} e^{b_j^* - \varepsilon_b - \varepsilon_p} \cdot \left(e^{\sigma\left(\frac{\gamma}{2k \max_{t \in S'(x)} |a_t^*|} - 2 \cdot \varepsilon_b - \varepsilon_p \right)} - 1 \right) \right\}, \tag{3.48}$$

where $S'(x) := \{j : a_j^* < 0, \Delta_j' > 0\}$ and $\Delta_j' := (\langle v_{-i}(x), v_j - v_0 \rangle - b_j^*) - \sigma(\langle v_{-i}^*(x), v_j^* - v_0^* \rangle - b_j^*)$.

Since $S(x) \cup S'(x) \subseteq [k]$, merging (3.46) and (3.48) finishes the proof of Lemma 3.5.9.

□

3.6 Conclusions and Future Work

In this paper, we analyzed when and how the representations generated by pre-trained large-scale language models can be used in downstream classification tasks.

We found two necessary conditions for guaranteeing the usefulness of these representations: the insensitivity of the downstream task to super-small probability words, and the underlying structure in the representations to handle the shift-invariance of softmax. We also provide a sufficient condition, i.e., the existence of an anchor vector, that can guarantee the representations from pre-trained language model to help with downstream tasks. We verify this existence empirically in various large language models and believe that this is an important reason why recent large-scale language models can adapt to different downstream tasks.

While our work showed the existence of the anchor vector, it remains unclear why this vector exists in most large language models. It might be related to the initialization, optimization, and structure of the neural networks, especially transformers, and it could also be related to the underlying structure of the training data. Digging deeper into this may reveal more fundamental properties of these models. Our analysis of downstream task focuses on classification tasks. Since these large networks perform well in various types of downstream tasks, another future direction would be to analyze other kinds of downstream tasks. Furthermore, we model the network before the softmax function as a black box, and further opening up this black box is important for deeper understanding of the structures in the learned representations from pre-training.

4

Understanding Structure of Hessian for Neural Networks

In the previous chapters, we assume that the optimization of neural networks is successful, i.e., we can learn a set of parameters for the model so that the loss function achieves a small value. Beginning from this chapter, we will focus on the optimization landscape of practical neural networks. We are going to study two heuristics that are frequently used in practice for training deep neural networks and provide some theoretical insights. In this chapter, we focus on neural network loss Hessian.

Hessian captures important properties of the deep neural network optimization landscape. Previous works have observed low rank structure in the Hessians of neural networks. In this chapter, we propose a decoupling conjecture that decomposes the layer-wise Hessians of a network as the Kronecker product of two smaller matrices. We can analyze the properties of these smaller matrices and prove the structure of top eigenspace random 2-layer networks. The decoupling conjecture has several other interesting implications – top eigenspaces for different models have surprisingly high

overlap, and top eigenvectors form low rank matrices when they are reshaped into the same shape as the corresponding weight matrix. All of these can be verified empirically for deeper networks. Finally, we use the structure of layer-wise Hessian to get better explicit generalization bounds for neural networks.

4.1 Introduction

The loss landscape for neural networks is crucial for understanding training and generalization. In this paper we focus on the structure of Hessians, which capture important properties of the loss landscape. For optimization, Hessian information is used explicitly in second order algorithms, and even for gradient-based algorithms properties of the Hessian are often leveraged in analysis (Sra et al., 2012). For generalization, the Hessian captures the local structure of the loss function near a local minimum, which is believed to be related to generalization gaps (Keskar et al., 2017).

Several previous results including Sagun et al. (2018); Papyan (2018) observed interesting structures in Hessians for neural networks – it often has around c large eigenvalues where c is the number of classes. In this paper we ask:

Why does the Hessian of neural networks have special structures in its top eigenspace?

A rigorous analysis of the Hessian structure would potentially allow us to understand what the top eigenspace of the Hessian depends on (e.g., the weight matrices or data distribution), as well as predicting the behavior of the Hessian when the architecture changes.

Towards this goal, we focus on the layer-wise Hessians in this paper. One difficulty in analyzing the layer-wise Hessian lies in its size – for a fully-connected layer with

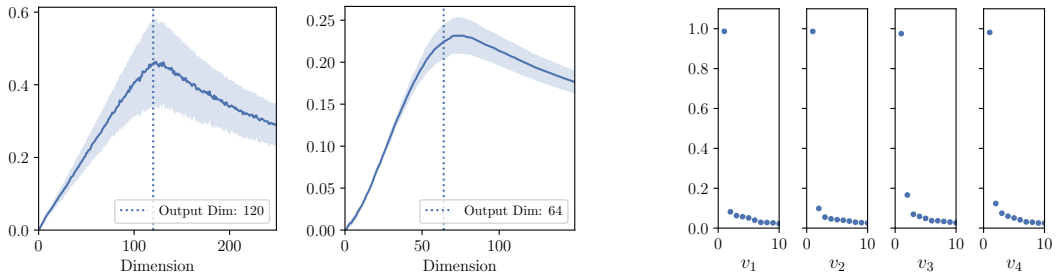
a $n \times n'$ weight matrix, the layer-wise Hessian is a $nn' \times nn'$ matrix. We propose a *decoupling conjecture* that approximates this matrix by the Kronecker product of two smaller matrices – a $n \times n$ input autocorrelation matrix and a $n' \times n'$ output Hessian matrix. We then study the properties of these two smaller matrices, which together with the decoupling conjecture give an explanation of why there are just a few large eigenvalues, as well as a heuristic formula to efficiently compute the top eigenspace. We prove the decoupling conjecture and structure of the output Hessian matrix for a simple model of 2-layer network. We then empirically verify that these results extend to much more general settings.

4.1.1 Outline

Understanding Hessian Structure using Kronecker Factorization. In Section 4.3 We first formalize a decoupling conjecture that states the layer-wise Hessian can be approximated by the Kronecker product of the output Hessian and input auto-correlation.

The auto-correlation of the input is often very close to a rank 1 matrix, because the inputs for most layers have a nonzero expectation. We show that when the input auto-correlation component is approximately rank 1, top eigenspace of the layerwise Hessian is very similar to that of the output Hessian. On the contrary, when inputs have mean 0 (e.g., when the model is trained with batch normalization), the input auto-correlation matrix is much farther from rank 1 and the layer-wise Hessian often does not have the same low rank structure.

In Section 4.4 we prove that in an over-parametrized two-layer neural network on random data, the output Hessian is approximately rank $c - 1$. Further, we can compute the top $c - 1$ eigenspace directly from weight matrices. We show a similar low rank result for the layer-wise Hessian.



(a) Overlap between dominate eigenspace of layer-wise Hessian at different minima for fc1:LeNet5 (**left**) with output dimension 120 and conv11:ResNet18-W64 (**right**) with output dimension 64.

(b) Top 10 singular values of the top 4 eigenvectors of the layer-wise Hessian of fc1:LeNet5 after reshaped as matrix.

FIGURE 4.1: Some interesting observations on the structure of layer-wise Hessians. The eigenspace overlap is defined in Definition 4.2.1 and the reshape operation is defined in Definition 4.2.2

Implication on the Structure of Top Eigenspace for Hessians. The decoupling conjecture, together with our characterizations of its two components, have surprising implications to the structure of top-eigenspace for layer-wise Hessians. Since the eigenvector of a Kronecker product is just the outer product of eigenvectors of its components, if we express the top eigenvectors of a layer-wise Hessian as a matrix with the same dimensions as the weight matrix, then the matrix is approximately rank 1. In Figure 4.1.a we show the singular values of several such reshaped eigenvectors. Another more surprising phenomenon considers the overlap between top eigenspaces for different models.

Consider two neural networks trained with different random initializations and potentially different hyper-parameters; their weights are usually nearly orthogonal. One might expect that the top eigenspace of their layer-wise Hessians are also very different. However, empirically one observe that the top eigenspace of the layer-wise Hessians have a very high overlap, and the overlap peaks at the dimension of the layer's output (see Figure 4.1(a)). This is a direct consequence of the Kronecker product and the fact that the input auto-correlation matrix is close to rank 1.

Applications. As a direct application of our results, in Section 4.6 we show that the Hessian structure can be used to improve the PAC-Bayes bound computed in Dziugaite and Roy (2017).

4.1.2 Related Works

Hessian-based analysis for neural networks (NNs). Hessian matrices for NNs reflect the second order information about the loss landscape, which is important in characterizing SGD dynamics (Jastrzebski et al., 2019) and related to generalization (Li et al., 2020), robustness to adversaries (Yao et al., 2018) and interpretation of NNs (Singla et al., 2019). People have empirically observed several interesting phenomena of the Hessian, e.g., the gradient during training converges to the top eigenspace of Hessian (Gur-Ari et al., 2018; Ghorbani et al., 2019), and the eigenspectrum of Hessian contains a "spike" which has about $c-1$ large eigenvalues and a continuous "bulk" (Sagun et al., 2016, 2018; Papyan, 2018). People have developed different frameworks to explain the low rank structure of the Hessians including hierarchical clustering of logit gradients (Papyan, 2019, 2020), independent Gaussian model for logit gradients (Fort and Ganguli, 2019), and Neural Tangent Kernel (Jacot et al., 2020). A distinguishing feature of this work is that we are able to characterize the top eigenspace of the Hessian directly by the weight matrices of the network.

Layer-wise Kronecker factorization (K-FAC) for training NNs. The idea of using Kronecker product to approximate Hessian-like matrices is not new. Heskes (2000) uses this idea to approximate Fisher Information Matrix (FIM). Martens and Grosse (2015) proposed Kronecker-factored approximate curvature which approximates the inverse of FIM using layer-wise Kronecker product. Kronecker factored eigenbasis has also been utilized in training (George et al., 2018). Our paper focuses on a dif-

ferent application with different matrix (Hessian vs. inverse FIM) and different ends of the spectrum (top vs. bottom eigenspace).

Theoretical Analysis for Hessians Eigenstructure. Karakida et al. (2019b) showed that the largest c eigenvalues of the FIM for a randomly initialized neural network are much larger than the others. Their results rely on the eigenvalue spectrum analysis in Karakida et al. (2019c,a), which assumes the weights used during forward propagation are drawn independently from the weights used in back propagation (Schoenholz et al., 2017). More recently, Singh et al. (2021) provided a Hessian rank formula for linear networks and Liao and Mahoney (2021) provided a characterization on the eigenspace structure of G-GLM models (including 1-layer NN). To our best knowledge, theoretical analysis on the Hessians of nonlinear deeper neural networks is still vacant.

PAC-Bayes generalization bounds. People have established generalization bounds for neural networks under PAC-Bayes framework by McAllester (1999). For neural networks, Dziugaite and Roy (2017) proposed the first non-vacuous generalization bound, which used PAC-Bayesian approach with optimization to bound the generalization error for a stochastic neural network.

4.2 Preliminaries and Notations

Basic Notations. In this paper, we generally follow the default notation suggested by Goodfellow et al. (2016). Additionally, for a matrix $\boldsymbol{M}$, let $\|\boldsymbol{M}\|_F$ denote its Frobenius norm and $\|\boldsymbol{M}\|$ denote its spectral norm. For two matrices $\boldsymbol{M} \in \mathbb{R}^{a_1 \times b_1}, \boldsymbol{N} \in \mathbb{R}^{a_2 \times b_2}$, let $\boldsymbol{M} \otimes \boldsymbol{N} \in \mathbb{R}^{(a_1 a_2) \times (b_1 b_2)}$ be their Kronecker product such that $[\boldsymbol{M} \otimes \boldsymbol{N}]_{(i_1-1)\times a_2 + i_2, (j_1-1)\times b_2 + j_2} = \boldsymbol{M}_{i_1,i_2} \boldsymbol{N}_{j_1,j_2}$.

Neural Networks. For a c-class classification problem with training samples $S = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$ where $(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathbb{R}^d \times \{0,1\}^c$ for all $i \in [N]$, assume S is i.i.d. sampled from the underlying data distribution $\mathcal{D}$. Consider an L-layer fully connected ReLU neural network $f_\theta : \mathbb{R}^d \to \mathbb{R}^c$. With $\sigma(x) = x\mathbf{1}_{x \geqslant 0}$ as the Rectified Linear Unit (ReLU) function, the output of this network is a series of logits $\boldsymbol{z} \in \mathbb{R}^c$ computed recursively as $\boldsymbol{z}^{(p)} := \boldsymbol{W}^{(p)}\boldsymbol{x}^{(p)} + \boldsymbol{b}^{(p)}$ and $\boldsymbol{x}^{(p)} := \sigma(\boldsymbol{z}^{(p)})$

Here we denote the input and output of the p-th layer as $\boldsymbol{x}^{(p)}$ and $\boldsymbol{z}^{(p)}$, and set $\boldsymbol{x}^{(1)} = \boldsymbol{x}$, $\boldsymbol{z} := f_\theta(\boldsymbol{x}) = \boldsymbol{z}^{(L)}$. We denote $\theta := (\boldsymbol{w}^{(1)}, \boldsymbol{b}^{(1)}, \boldsymbol{w}^{(2)}, \boldsymbol{b}^{(2)}, \cdots, \boldsymbol{w}^{(L)}, \boldsymbol{b}^{(L)}) \in \mathbb{R}^P$ the parameters of the network. For the i-th layer, $\boldsymbol{w}^{(i)}$ is the flattened weight matrix $\boldsymbol{W}^{(i)}$ and $\boldsymbol{b}^{(i)}$ is its corresponding bias vector. For convolutional networks, a similar framework is introduced in Section C.1.2.

For a single input $\boldsymbol{x} \in \mathbb{R}^d$ with one-hot label $\boldsymbol{y}$ and logit output $\boldsymbol{z}$, let $n^{(p)}$ and $m^{(p)}$ be the lengths of $\boldsymbol{x}^{(p)}$ and $\boldsymbol{z}^{(p)}$. For convolutional layers, we consider the number of output channels as $m^{(p)}$ and width of unfolded input as $n^{(p)}$. Note that $\boldsymbol{x}^{(1)} = \boldsymbol{x}, \boldsymbol{z}^{(L)} = \boldsymbol{z} = f_\theta(\boldsymbol{x})$. We denote $\boldsymbol{p} := \mathrm{softmax}(\boldsymbol{z}) = e^{\boldsymbol{z}} / \sum_{i=1}^c e^{\boldsymbol{z}_i}$ as the output confidence. With the cross-entropy loss function $\ell(\boldsymbol{p}, \boldsymbol{y}) = -\sum_{i=1}^c \boldsymbol{y}_i \log(\boldsymbol{p}_i) \in \mathbb{R}^+$, the training process optimizes parameter θ to minimize the empirical training loss $\mathcal{L}(\theta) := \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\in S}\left[\ell\left(\boldsymbol{z}, \boldsymbol{y}\right)\right]$.

Hessians. Fixing the parameter θ, we use $\boldsymbol{H}_\ell(\boldsymbol{v}, \boldsymbol{x}) = \nabla^2_{\boldsymbol{v}} \ell(f_\theta(\boldsymbol{x}), \boldsymbol{y}) = \nabla^2_{\boldsymbol{v}} \ell(\boldsymbol{z}, \boldsymbol{y})$ to denote the Hessian of some vector $\boldsymbol{v}$ with respect to scalar loss function ℓ at input $\boldsymbol{x}$. Note that $\boldsymbol{v}$ can be any vector. For example, the full parameter Hessian is $\boldsymbol{H}_\ell(\theta, \boldsymbol{x})$ where we take $\boldsymbol{v} = \theta$, and the layer-wise weight Hessian of the p-th layer is $\boldsymbol{H}_\ell(\boldsymbol{w}^{(p)}, \boldsymbol{x})$ where we take $\boldsymbol{v} = \boldsymbol{w}^{(p)}$.

For simplicity, define $\mathbb{E}$ as the empirical expectation operator over the training

sample S unless explicitly stated otherwise. We mainly focus on the layer-wise weight Hessians $\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)}) = \mathbb{E}[\boldsymbol{H}_{\ell}(\boldsymbol{w}^{(p)}, \boldsymbol{x})]$ with respect to loss, which are diagonal blocks in the full Hessian $\boldsymbol{H}_{\mathcal{L}}(\theta) = \mathbb{E}[\boldsymbol{H}_{\ell}(\theta, \boldsymbol{x})]$ corresponding to the cross terms between the weight coefficients of the same layer. We define $\boldsymbol{M}_{\boldsymbol{x}}^{(p)} := \boldsymbol{H}_{\ell}(\boldsymbol{z}^{(p)}, \boldsymbol{x})$ as the Hessian of output $\boldsymbol{z}^{(p)}$ with respect to empirical loss. With the notations defined above, we have the p-th layer-wise Hessian for a single input as

$$\boldsymbol{H}_{\ell}(\boldsymbol{w}^{(p)}, \boldsymbol{x}) = \nabla^2_{\boldsymbol{w}^{(p)}} \ell(\boldsymbol{z}, \boldsymbol{y}) = \boldsymbol{M}_{\boldsymbol{x}}^{(p)} \otimes (\boldsymbol{x}^{(p)} \boldsymbol{x}^{(p)T}). \tag{4.1}$$

It follows that

$$\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)}) = \mathbb{E}\left[\boldsymbol{M}_{\boldsymbol{x}}^{(p)} \otimes \boldsymbol{x}^{(p)} \boldsymbol{x}^{(p)T}\right] = \mathbb{E}\left[\boldsymbol{M} \otimes \boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}\right]. \tag{4.2}$$

The subscription $\boldsymbol{x}$ and the superscription (p) will be omitted when there is no confusion, as our analysis primarily focuses on the same layer unless otherwise stated. We also define subspace overlap and layer-wise eigenvector matricization for our analysis.

Definition 4.2.1 (Subspace Overlap)**.** For k-dimensional subspaces $\boldsymbol{U}, \boldsymbol{V}$ in $\mathbb{R}^d$ $(d \geqslant k)$ where the basis vectors $\boldsymbol{u}_i$'s and $\boldsymbol{v}_i$'s are column vectors, with $\boldsymbol{\phi}$ as the size k vector of canonical angles between $\boldsymbol{U}$ and $\boldsymbol{V}$, we define the subspace overlap of $\boldsymbol{U}$ and $\boldsymbol{V}$ as $\text{Overlap}(\boldsymbol{U}, \boldsymbol{V}) := \|\boldsymbol{U}^{\mathsf{T}}\boldsymbol{V}\|_F^2/k = \|\cos\boldsymbol{\phi}\|_2^2/k$.

Definition 4.2.2 (Layer-wise Eigenvector Matricization)**.** Consider a layer with input dimension n and output dimension m. For an eigenvector $\boldsymbol{h} \in \mathbb{R}^{mn}$ of its layer-wise Hessian, the matricized form of $\boldsymbol{h}$ is $\text{Mat}(\boldsymbol{h}) \in \mathbb{R}^{m \times n}$ where $\text{Mat}(\boldsymbol{h})_{i,j} = \boldsymbol{h}_{(i-1)m+j}$.

4.3 Decoupling Conjecture and Implications on the Structures of Hessian

The fact that layer-wise Hessian for a single sample can be decomposed into Kronecker product of two components naturally leads to the following informal conjecture:

Conjecture (Decoupling Conjecture). The layer-wise Hessian can be approximated by a Kronecker product of the expectation of its two components, that is

$$\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)}) = \mathbb{E}[\boldsymbol{M} \otimes \boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}] \approx \mathbb{E}[\boldsymbol{M}] \otimes \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}]. \tag{4.3}$$

More specifically, we conjecture that $\frac{\|\mathbb{E}[\boldsymbol{M}]\otimes\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}]-\mathbb{E}[\boldsymbol{M}\otimes\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}]\|}{\|\mathbb{E}[\boldsymbol{M}\otimes\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}]\|} \leqslant \epsilon$, where ϵ is a small constant.

Note that this conjecture is certainly true when $\boldsymbol{M}$ and $\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}$ are approximately statistically independent. One immediate implication is that the top eigenvalues and eigenspace of $\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)})$ is close to those of $\mathbb{E}[\boldsymbol{M}] \otimes \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}]$. In Section 4.4 we prove that the eigenspaces are indeed close for a simple setting, and in Section 4.5.1 we show that this conjecture is empirically true in practice.

Assuming the decoupling conjecture, we can analyze the layer-wise Hessian by analyzing the two components separately. Note that $\mathbb{E}[\boldsymbol{M}]$ is the Hessian of the layer-wise output with respect to empirical loss, and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}]$ is the auto-correlation matrix of the layer-wise inputs. For simplicity we call $\mathbb{E}[\boldsymbol{M}]$ the output Hessian and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}]$ the input auto-correlation. For convolutional layers we can a similar factorization $\mathbb{E}[\boldsymbol{M}] \otimes \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}]$ for the layer-wise Hessian, but with a different $\boldsymbol{M}$ motivated by Grosse and Martens (2016). (See Section C.1.2) We note that the off-diagonal blocks of the full Hessian can also be decomposed similarly, which in turn allows us to approximate the eigenvalues and eigenvectors of the full parameter

Hessian. The details of this approximation is stated in Section C.3.

4.3.1 Structure of Input Auto-Correlation Matrix $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ and Output Hessian $\mathbb{E}[\boldsymbol{M}]$

For the auto-correlation matrix, one can decompose it as $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{x}]^\top + \mathrm{Var}[\boldsymbol{x}]$. A key observation is that the input $\boldsymbol{x}$ for most layers are outputs of a ReLU, hence it is nonnegative. For large networks the mean component $\mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{x}]^\top$ will dominate the variance, making $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ approximately rank-1 with top eigenvector being very close to $\mathbb{E}[\boldsymbol{x}]$. We empirically verified this phenomenon on a variety of networks and datasets (see Section C.6.1).

For the output Hessian, we observe that $\mathbb{E}[\boldsymbol{M}]$ is approximately rank $c-1$ (with $c-1$ significantly large eigenvalues) in most cases. In Section 4.4, we show this is indeed the case in a simplified setting, and give a formula for computing the top $c-1$ eigenspace using rows of weight matrices.

4.3.2 Implications on the Eigenspectrum and Eigenvectors of Layer-Wise Hessian

The eigenvectors of a Kronecker product is the tensor product of eigenvectors of its components. As a result, let $\boldsymbol{h}_i$ be the i-th eigenvector of a layer-wise Hessian $\boldsymbol{H}$, if we matricize it as defined in Definition 4.2.2, $\mathrm{Mat}(\boldsymbol{h}_i)$ would be approximately rank 1. Since $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ is close to rank 1, by the decoupling conjecture, the top eigenvalues of layer-wise Hessian can be approximated as the top eigenvalues of $\mathbb{E}[\boldsymbol{M}]$ multiplied by the first eigenvalue of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$. The low rank structure of the layer-wise Hessian $\boldsymbol{H}$ is due to the low rank structure of $\mathbb{E}[\boldsymbol{M}]$.

Another implication is related to eigenspace overlap for different models. Even though the output Hessians of two randomly trained models may be very different, the top eigenspace of the Hessian will be close to $\mathbb{E}[\boldsymbol{x}] \otimes I$, so the top eigenspace of the two models will have a high overlap that peaks at the output dimension. See

Section 4.5.3 for more details.

4.4 Hessian Structure for Infinite Width Two-Layer ReLU Neural Network

In this section, we show that for a simple setting of 2-layer networks, the layer-wise parameter Hessian has $c-1$ large eigenvalues and its top $c-1$ eigenspace is close to the top $c-1$ eigenspace of the Kronecker product approximation.

Problem Setting and Notations Let bold non-italic letters such as $\mathbf{v}, \mathbf{M}$ denote random vectors (lowercase) and matrices (uppercase). Consider a two layer fully connected ReLU activated neural network with input dimension d, hidden layer dimension n and output dimension c. In particular, let $d = n^{1+\alpha}$ for some constant $\alpha > 0$. Let the network has positive input from a rectified Gaussian $\mathbf{x} \sim \mathcal{N}^{\mathrm{R}}(0, \boldsymbol{I}_d)$ where every entry is identically distributed as $\max\{\widehat{\mathbf{x}}, 0\}$ for $\widehat{\mathbf{x}} \sim \mathcal{N}(0,1)$. Let $\boldsymbol{W}^{(1)} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{W}^{(2)} \in \mathbb{R}^{c\times n}$ be the weight matrices. In this problem we consider a random Gaussian initialization that $\boldsymbol{W}^{(1)} \sim \mathcal{N}(0, \frac{1}{d}\boldsymbol{I}_{dn})$ and $\boldsymbol{W}^{(2)} \sim \mathcal{N}(0, \frac{1}{n}\boldsymbol{I}_{nc})$. Both weight matrices has expected row norm of 1. Let the loss objective be cross entropy ℓ. Training labels are irrelevant as they are independent from the Hessian at initialization.

Denote the output of the first and second layer as $\mathbf{y}$ and $\mathbf{z}$ respectively. We have $\mathbf{y} = \sigma(\boldsymbol{W}^{(1)}\mathbf{x})$ and $\mathbf{z} = \boldsymbol{W}^{(2)}\mathbf{y}$. Here σ is the element-wise ReLU function. Let $\mathbf{D} \triangleq \operatorname{diag}(\mathbb{I}[\mathbf{y} \geqslant 0]) \in \mathbb{R}^{n\times n}$ denote the 0/1 diagonal matrix representing the activation of σ that $\mathbf{y} = \mathbf{D}\boldsymbol{W}^{(1)}\mathbf{x}$. Let $\mathbf{p} = \operatorname{softmax}(\mathbf{z})$ and let $\mathbf{A} \triangleq \operatorname{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^{\mathsf{T}}$. Note $\mathbf{A}$ is rank $c-1$ with the null space of the all one vector. We give full details about our settings in Section C.2.1. By simple matrix calculus (see Section C.1.1),

the output Hessian of $\boldsymbol{M}^{(1)}$ and the full layer-wise Hessian has closed-form

$$\boldsymbol{M}^{(1)} = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{N}^{\mathrm{R}}(0,\boldsymbol{I}_d)}\left[\mathbf{D}\boldsymbol{W}^{(2)\mathsf{T}}\mathbf{A}\boldsymbol{W}^{(2)}\mathbf{D}\right], \boldsymbol{H}^{(1)} = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{N}^{\mathrm{R}}(0,\boldsymbol{I}_d)}\left[\mathbf{D}\boldsymbol{W}^{(2)\mathsf{T}}\mathbf{A}\boldsymbol{W}^{(2)}\mathbf{D}\otimes\mathbf{x}\mathbf{x}^{\mathsf{T}}\right]. \tag{4.4}$$

Following the decoupling conjecture, the Kronecker approximation of the layer-wise Hessian is

$$\widehat{\boldsymbol{H}}^{(1)} \triangleq \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{N}^{\mathrm{R}}(0,\boldsymbol{I}_d)}\left[\mathbf{D}\boldsymbol{W}^{(2)\mathsf{T}}\mathbf{A}\boldsymbol{W}^{(2)}\mathbf{D}\right]\otimes\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{N}^{\mathrm{R}}(0,\boldsymbol{I}_d)}\left[\mathbf{x}\mathbf{x}^{\mathsf{T}}\right]. \tag{4.5}$$

Since we are always taking the expectation over the input $\mathbf{x}$, we will neglect the subscript and use $\mathbb{E}$ for expectation. Now we are ready to state our main theorem.

Theorem 4.4.1. *For an infinite width two-layer ReLU activated neural network with Gaussian initialization as defined above, let V_1 and V_2 be the top $c-1$ eigenspaces of $\boldsymbol{H}^{(1)}$ and $\widehat{\boldsymbol{H}}^{(1)}$ respectively, for all $\epsilon > 0$,*

$$\lim_{n\to\infty}\Pr_{\boldsymbol{W}^{(1)}\sim\mathcal{N}(0,\frac{1}{d}\boldsymbol{I}_{nd}),\boldsymbol{W}^{(2)}\sim\mathcal{N}(0,\frac{1}{n}\boldsymbol{I}_{cn})}\left[\mathrm{Overlap}\left(V_1,V_2\right) > 1-\epsilon\right] = 1.$$

Moreover $\boldsymbol{H}^{(1)}$ has $c-1$ large eigenvalues that,

$$\lim_{n\to\infty}\Pr_{\boldsymbol{W}^{(1)}\sim\mathcal{N}(0,\frac{1}{d}\boldsymbol{I}_{nd}),\boldsymbol{W}^{(2)}\sim\mathcal{N}(0,\frac{1}{n}\boldsymbol{I}_{cn})}\left[\left(\frac{\lambda_c(\boldsymbol{H}^{(1)})}{\lambda_{c-1}(\boldsymbol{H}^{(1)})}\bigg|_{\boldsymbol{W}^{(1)},\boldsymbol{W}^{(2)}}\right) < \epsilon\right] = 1. \tag{4.6}$$

Instead of directly working on the layer-wise Hessian, we first show a similar theorem for the output Hessian $\boldsymbol{M}^{(1)}$. We will then show that the proof technique of the following theorem can be easily generalized to prove our main theorem.

Theorem 4.4.2. *For the same network as in Theorem 4.4.1, let*

$$\boldsymbol{M}^* \triangleq \mathbb{E}\left[\mathbf{D}'\boldsymbol{W}^{(2)\mathsf{T}}\mathbf{A}\boldsymbol{W}^{(2)}\mathbf{D}'\right]$$

where $\mathbf{D}'$ *is an independent copy of* $\mathbf{D}$ *and is independent of* $\mathbf{A}$*. Let* S_1 *and* S_2 *be the top* $c-1$ *eigenspaces of* $\boldsymbol{M}^{(1)}$ *and* $\boldsymbol{M}^*$ *respectively,* S_2 *is approximately* $\mathcal{R}\{\boldsymbol{W}_i\}_{i=1}^c \backslash \{\boldsymbol{1}^\top \boldsymbol{W}\}$ *where* $\mathcal{R}$ *is the row span, and for all* $\epsilon > 0$*,*

$$\lim_{n\to\infty} \Pr_{\boldsymbol{W}^{(1)}\sim\mathcal{N}(0,\frac{1}{d}\boldsymbol{I}_{nd}),\boldsymbol{W}^{(2)}\sim\mathcal{N}(0,\frac{1}{n}\boldsymbol{I}_{cn})} [\text{Overlap}\,(S_1, S_2) > 1-\epsilon] = 1.$$

Moreover, $\boldsymbol{M}$ *has* $c-1$ *large eigenvalues that*

$$\lim_{n\to\infty} \Pr_{\boldsymbol{W}^{(1)}\sim\mathcal{N}(0,\frac{1}{d}\boldsymbol{I}_{nd}),\boldsymbol{W}^{(2)}\sim\mathcal{N}(0,\frac{1}{n}\boldsymbol{I}_{cn})} \left[\left(\frac{\lambda_c(\boldsymbol{M}^{(1)})}{\lambda_{c-1}(\boldsymbol{M}^{(1)})}\bigg|_{\boldsymbol{W}^{(1)},\boldsymbol{W}^{(2)}} \right) < \epsilon \right] = 1. \tag{4.7}$$

Proof Sketch for Theorem 4.4.2 For simplicity of notations, in this section we will use $\boldsymbol{W}$ to denote $\boldsymbol{W}^{(2)}$ and $\boldsymbol{M}$ to denote $\boldsymbol{M}^{(1)}$ unless specified otherwise. Our proof of Theorem 4.4.2 mainly consists of three parts. First we analyze the structure of $\boldsymbol{M}^*$ and show that it is approximately rank $c-1$. Then we show that $\boldsymbol{M}^*$ and $\boldsymbol{M}$ are roughly equivalent via an approximate independence between $\mathbf{D}$ and $\mathbf{A}$. Finally, by projecting both $\boldsymbol{M}$ and $\boldsymbol{M}^*$ onto a $c \times c$ matrix using $\boldsymbol{W}$, we can apply the approximate independence and prove that the top $c-1$ eigenspace of $\boldsymbol{M}^*$ is approximately that of $\boldsymbol{M}$, which concludes the proof.

(1) Structure of $\boldsymbol{M}^*$ When $n \to \infty$, the output of the second layer $\mathbf{y}$ converges to a multivariate Gaussian (Lemma C.2.11), hence we can consider each diagonal entry of $\mathbf{D}$ as a $p = \frac{1}{2}$ Bernoulli random variable. Since we assumed that $\mathbf{D}'$ and $\mathbf{A}$ are independent, by some simple calculation,

$$\boldsymbol{M}^* = \frac{1}{4}\left(\boldsymbol{W}^\top \mathbb{E}[\mathbf{A}]\boldsymbol{W} + \text{diag}(\boldsymbol{W}^\top \mathbb{E}[\mathbf{A}]\boldsymbol{W})\right). \tag{4.8}$$

Here $\mathbb{E}[\mathbf{A}]$ is rank $c-1$ with the $(c-1)$-th eigenvalue bounded below from 0 (Lemma C.2.15). Since the two terms in the sum has the same trace while $\boldsymbol{W}^\top \mathbb{E}[\mathbf{A}]\boldsymbol{W}$ is rank $c-1$ compared to rank n of $\text{diag}(\boldsymbol{W}^\top \mathbb{E}[\mathbf{A}]\boldsymbol{W})$, we can show that the top eigenspace is dominated by the eigenspace of $\boldsymbol{W}^\top \mathbb{E}[\mathbf{A}]\boldsymbol{W}$, which is approximately $\mathcal{R}\{\boldsymbol{W}_i\}_{i=1}^c \backslash \{\mathbf{1}^\top \boldsymbol{W}\}$.

(2) Approximate Independence Between $\mathbf{A}$ *and* $\mathbf{D}$ Intuitively, if $\mathbf{D}$ and $\mathbf{A}$ are independent, then $\boldsymbol{M} = \boldsymbol{M}^*$. However, this is clearly not true - if the activations align with a row of $\boldsymbol{W}$ then the corresponding output is going to be large, which changes $\mathbf{A}$ significantly. To address this problem, we observe that the formula for $\boldsymbol{M}$ is only of degree 2 in $\mathbf{D}$, so one can focus on conditioning on two of the activations – a negligible fraction in the limit. More precisely, if one expand out the expression of each element squared in $\boldsymbol{M}$, it is an homogeneous polynomial of the form $p(\mathbf{A}, \mathbf{D}, \bar{\mathbf{A}}, \bar{\mathbf{D}}) = \sum_{i,j,k,l=1}^{c} \sum_{p,q=1}^{n} c_{ijklpq} \mathbf{A}_{ij} \bar{\mathbf{A}}_{kl} \mathbf{D}_{pp} \bar{\mathbf{D}}_{qq}$, where $(\bar{\mathbf{A}}, \bar{\mathbf{D}})$ are independent copies of $(\mathbf{A}, \mathbf{D})$. The same element squared in $\boldsymbol{M}^*$ is just going to be $p(\mathbf{A}, \mathbf{D}', \bar{\mathbf{A}}, \bar{\mathbf{D}}')$. By nice properties of the Gaussian initialized weight matrix, we show that as $n \to \infty$, $\mathbf{A}$ is invariant when conditioning on two entries of $\mathbf{D}$ (Lemma C.2.13). Therefore, in the limit we have $\lim_{n\to\infty} \mathbb{E}\left[p(\mathbf{A}, \mathbf{D}, \bar{\mathbf{A}}, \bar{\mathbf{D}})\right] = \mathbb{E}\left[p(\mathbf{A}, \mathbf{D}', \bar{\mathbf{A}}, \bar{\mathbf{D}}')\right]$ (detailed proof in Appendix).

(3) Equivalence between $\boldsymbol{M}^*$ *and* $\boldsymbol{M}$ Since the size of $\boldsymbol{M}$ also goes to infinity as we take the limit on n, it is technically difficult to directly compare their eigenspaces. In this case we utilize the fact that $\boldsymbol{W}$ has approximately orthogonal rows, and project $\boldsymbol{M}$ onto $\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top$. In particular, by expanding out the Frobenious norms as polynomials and bounding the ℓ_1 norm of the coefficients, using Lemma C.2.13 we are able to show that $\|\boldsymbol{M}\|_F^2 \approx \left\|\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\right\|_F^2 \approx \left\|\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\top\right\|_F^2 \approx \|\boldsymbol{M}^*\|_F^2$ (Lemma C.2.18- Lemma C.2.22). This result tells us that the projection does not

lose information, and hence indirectly gives us the dominating eigenspace of $\boldsymbol{M}$. This concludes our proof for Theorem 4.4.2

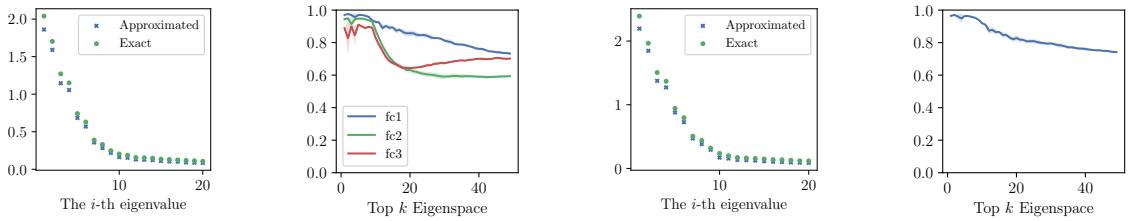
Proving Theorem 4.4.1 and Beyond To prove Theorem 4.4.1, we use a very similar strategy. We consider a re-scaled Hessian $\breve{\boldsymbol{H}} \triangleq \frac{1}{d}\boldsymbol{H}$ and show that in the independent setting $\breve{\boldsymbol{H}}^* = \frac{1}{d}\mathbb{E}[\mathbf{D}'\boldsymbol{W}\mathbf{A}\boldsymbol{W}\mathbf{D}' \otimes \mathbf{x}''\mathbf{x}''^{\mathsf{T}}] = \boldsymbol{M}^* \otimes \frac{1}{d}\mathbb{E}[\mathbf{x}''\mathbf{x}''^{\mathsf{T}}]$. We then generalize the conditioning technique to involve conditioning on two entries of x.

4.5 Empirical Observation and Verification

In this section, we present some empirical observations that either verifies, or are induced by the decoupling conjecture. We conduct experiments on the CIFAR-10, CIFAR-100 (Krizhevsky and Hinton, 2009), and MNIST (LeCun et al., 1998) datasets as well as their random labeled versions, namely MNIST-R and CIFAR10-R. We used different fully connected (fc) networks (a fc network with m hidden layers and n neurons each hidden layer is denoted as F-n^m), several variations of LeNet (LeCun et al., 1998), VGG11 (Simonyan and Zisserman, 2015), and ResNet18 (He et al., 2016b). We use "layer:network" to denote a layer of a particular network. For example, conv2:LeNet5 refers to the second convolutional layer in LeNet5. More empirical results are included in Section C.6.

4.5.1 Kronecker Approximation of Layer-Wise Hessian and Full Hessian

To verify the decoupling conjecture in practical settings, we compare the top eigenvalues and eigenspaces of the approximated Hessian and the true Hessian. We use subspace overlap (Definition 4.2.1) to measure the similarity between top eigenspaces. As shown in Figure 4.2, this approximation works reasonably well on the top eigenspace.



(a) Top eigenvalues of layer-wise Hessian of fc1

(b) Top eigenspace of layer-wise Hessians

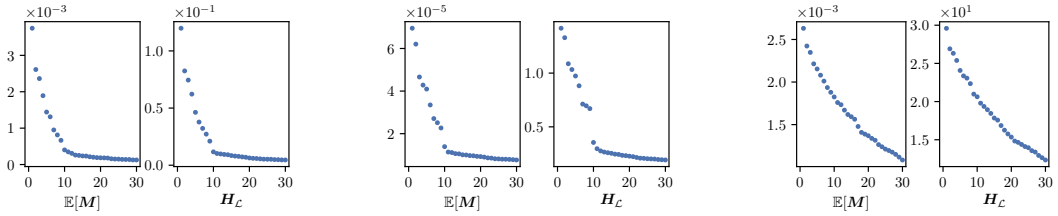
(c) Top eigenvalues of the full Hessian

(d) Top eigenspace of the full Hessian

FIGURE 4.2: Comparison between the approximated and true layer-wise Hessian of F-200[2].

4.5.2 Low Rank Structure of $\mathbb{E}[\boldsymbol{M}]$ *and* $\boldsymbol{H}$

Another way to empirically verify the decoupling conjecture is to show the similarity between the outliers in eigenspectrum of the layer-wise Hessian $\mathbb{E}[\boldsymbol{M}]$ and the output Hessian $\boldsymbol{H}_{\mathcal{L}}$. Figure 4.3 shows the similarity of eigenvalue spectrum between $\mathbb{E}[\boldsymbol{M}]$ and layer-wise Hessians in different situations, which agrees with our prediction. For (a) and (b) we are also seeing the eigengap at $c-1$, which is consistent with our analysis and previous observations (Sagun et al., 2018; Papyan, 2019). However, the eigengap does not appear at minimum for random labeled data with a under-parameterized network, meaning that our theory may not generalize to all settings.



(a) fc1:LeNet5 at initialization (CIFAR10).

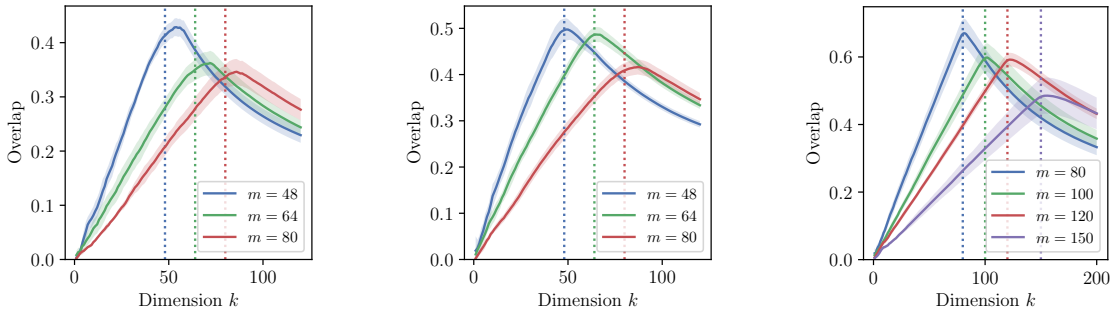
(b) fc1:LeNet5 at minimum (CIFAR10).

(c) fc1:LeNet5 at minimum (CIFAR10-R).

FIGURE 4.3: Eigenspectrum of the layer-wise output Hessian $\mathbb{E}[\boldsymbol{M}]$ and the layer-wise weight Hessian $\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)})$. The vertical axes denote the eigenvalues. Similarity between the two eigenspectra is a direct consequence of a low rank $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T]$ and the decoupling conjecture.

4.5.3 Eigenspace Overlap of Different Models

Apart from the phenomena that are direct consequences of the decoupling conjecture, we observe another nontrivial phenomenon involving different minima. Consider models with the same structure, trained on the same dataset, but using different random initializations, despite no obvious correlation between their parameters, we observe surpisingly high overlap between the dominating eigenspace of some of their layer-wise Hessians.



(a) conv12:ResNet18 (CIFAR100) with 48/64/80 output channels

(b) conv6:VGG11 (CIFAR100) with 48/64/80 output channels

(c) fc1:LeNet5 (CIFAR10) with 80/100/120/150 output neurons

FIGURE 4.4: Overlap between the top k dominating eigenspace of different independently trained models. The overlap peaks at the output dimension m. The eigenspace overlap is defined in Definition 4.2.1.

It turns out that the nontrivial overlap is also a consequence of the decoupling conjecture, which arises when the output Hessian and autocorrelation are related in the following way: When the small eigenvalues of $\mathbb{E}[\boldsymbol{M}] \in \mathbb{R}^{m \times m}$ approaches 0 slower than the small eigenvalues of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$, the top m eigenspace will then be approximately spanned by $\boldsymbol{I}_m \otimes \mathbb{E}[\boldsymbol{x}]^\mathsf{T}$ by the decoupling conjecture. Now suppose we have two different models with $\widehat{\mathbb{E}}[\boldsymbol{x}]_1$ and $\widehat{\mathbb{E}}[\boldsymbol{x}]_2$ respectively. Their top-m eigenspaces are approximately $\boldsymbol{I}_m \otimes \widehat{\mathbb{E}}[\boldsymbol{x}]_1$ and $\boldsymbol{I}_m \otimes \widehat{\mathbb{E}}[\boldsymbol{x}]_2$. Thus the overlap at dimension m is

approximately $(\widehat{\mathbb{E}}[\boldsymbol{x}]_1^\intercal \widehat{\mathbb{E}}[\boldsymbol{x}]_2)^2$, which is large since $\widehat{\mathbb{E}}[\boldsymbol{x}]_1$ and $\widehat{\mathbb{E}}[\boldsymbol{x}]_2$ are the same for the input layer and all non-negative for other layers. While this particular relation between $\mathbb{E}[\boldsymbol{M}]$ and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\intercal]$ are true in many shallow networks and in later layers of deeper networks, they are not satisfied for earlier layers of deeper networks. In Section C.7.2 we explain how one can still understand the overlap using correspondence matrices when the above simplified argument does not hold.

4.6 Tighter PAC-Bayes Bound with Hessian Information

The PAC-Bayes bound is a commonly used bound for the generalization gap of neural networks. In this section we show how we can obtain tighter PAC-Bayes bounds using the Kronecker approximation of Hessian eigenbasis.

Theorem 4.6.1 (PAC-Bayes Bound)**.** *(McAllester, 1999; Langford and Seeger, 2001) With the hypothesis space $\mathcal{H}$ parametrized by model parameters. For any prior distribution P in $\mathcal{H}$ that is chosen independently from the training set S, and any posterior distribution Q in $\mathcal{H}$ whose choice may inference S, with probability $1-\delta$, $\mathrm{KL}\left(\widehat{e}(Q)||e(Q)\right) \leqslant \frac{1}{|S|-1}\left[\mathrm{KL}(Q||P) + \log\frac{|S|}{\delta}\right]$. Where $e(Q)$ is the expected classification error for the posterior over the underlying data distribution and $\widehat{e}(Q)$ is the classification error for the posterior over the training set.*

Intuitively, if one can find a posterior Q that has low loss on the training set, and is close to the prior P, then the generalization error on Q must be small. Dziugaite and Roy (2017) uses optimization techniques to find an optimal posterior in the family of Gaussians with diagonal covariance. They showed that the bound can be nonvacuous for several neural network models.

We follow Dziugaite and Roy (2017) to set the prior P to be a multi-variant Gaussian. The covariance is invariant with respect to the change of basis since it

is a multiple of identity. Thus, For the posterior, when the variance in one direction is larger, the distance with the prior decreases; however this also has the risk of increasing the empirical loss over the posterior. In general, one would expect the variance to be larger along a flatter direction in the loss landscape and smaller along a sharper direction. However, since the covariance matrix of Q is fixed to be diagonal in Dziugaite and Roy (2017), the search of optimal deviation happens in standard basis vectors which are not aligned with the local loss landscape. Using the Kronecker factorization as in Equation 4.3, we can approximate the layer-wise Hessian's eigenspace. We set Q to be a Gaussian whose covariance is diagonal in the approximated eigenbasis of the layer-wise Hessians. Under this posterior change of basis, we can obtain tighter bounds compared to Dziugaite and Roy (2017). In our experiments, the final posterior variance $\boldsymbol{s}'$ is smaller along the direction of eigenvectors with larger eigenvalues (see Figure C.27). This agrees with our presumption that the alignment of sharp and flat directions will result in a better optimized posterior Q and thus a tighter bound on classification error.

Detailed algorithm description, experiment results, and plots are shown in Section C.8.

Table 4.1: Optimized PAC-Bayes bounds using different methods. T-n^m and R-n^m represents network F-n^m trained with true/random labels. TestEr. gives the empirical generalization gap. Base represents the bound given by Dziugaite and Roy (2017). Ours represents the bound we get.

Model	T-600	T-1200	T-300^2	T-600^2	R-600	T-600_{10}	T-200^2_{10}
TestEr.	0.015	0.016	0.015	0.015	0.493	0.018	0.021
Base	0.154	0.175	0.169	0.192	0.605	0.287	0.417
Ours	**0.120**	**0.142**	**0.125**	**0.146**	**0.568**	**0.213**	**0.215**

4.7 Limitations and Conclusions

In this paper we proposed the decoupling conjecture which helps in understanding many different structures for the top eigenspace of layer-wise Hessian. Our theory only applies to the initialization for a 2-layer network. How the property can be maintained throughout training is a major open problem. However, the implications of the decoupling conjecture can be verified empirically. Having such a conjecture allows us to predict how the structure of the Hessian changes based on architecture/training method (such as batch normalization), and has potential applications in understanding training and generalization (as we demonstrated by the new generalization bounds in Section 4.6). We hope this work would be a starting point towards formally proving the structures of neural network Hessians.

5

Learning-to-Learn for Tuning the Step Size

In this chapter, we study learning-to-learn, which is a commonly-used heuristic for choosing the parameters for the training algorithms of deep neural networks. These parameters can have large influence on the performance of the trained model, but the selection of a proper set of parameters usually requires trial-and-error approaches. Learning-to-learn formalizes the selection of these parameters as a machine learning problem and runs meta-gradient descent on a meta-objective to learn the parameters automatically. To understand and provide theoretical insights for learning-to-learn, we in this chapter work in a simple setting – choosing the step size for a quadratic objective. Under this setting, we give meta-optimization guarantees for the learning-to-learn approach.

Our results show that the naïve objective suffers from meta-gradient explosion or vanishing problem. Although there is a way to design the meta-objective so that the meta-gradient remains polynomially bounded, computing the meta-gradient directly using backpropagation leads to numerical issues. We also characterize when it is necessary to compute the meta-objective on a separate validation set to ensure the

generalization performance of the learned optimizer. Finally, we verify our results empirically and show that a similar phenomenon appears even for more complicated learned optimizers parametrized by neural networks.

5.1 Introduction

Choosing the right optimization algorithm and related hyper-parameters is important for training a deep neural network. Even for simple algorithms like gradient descent and stochastic gradient descent, choosing a good step size can be important to the convergence speed and generalization performance. Empirically, the parameters are often chosen based on past experiences or grid search. Recently, Maclaurin et al. (2015) considered the idea of tuning these parameters by optimization—that is, consider a meta-optimization problem where the goal is to find the best parameters for an optimizer. A series of works (e.g., Andrychowicz et al. (2016); Wichrowska et al. (2017)) extended such ideas and parametrized the set of optimizers by neural networks.

Although this approach has shown empirical success, there are very few theoretical guarantees for learned optimizers. Gupta and Roughgarden (2017) gave sample complexity bounds for tuning the step size, but they did not address how one can find the learned optimizer efficiently. In practice, the meta-optimization problem is often solved by meta-gradient descent—define a meta-objective function based on the trajectory that the optimizer generates, and then compute the meta-gradient using back-propagation (Franceschi et al., 2017). The optimization for meta-parameters is usually a nonconvex problem, therefore it is unclear why simple meta-gradient descent would find an optimal solution.

In this paper we consider using learning-to-learn approach to tune the step size

of standard gradient descent/stochastic gradient descent algorithm. Even in this simple setting, many of the challenges still remain and we can get better learned optimizers by choosing the right meta-objective function. Though our results are proved only in the simple setting, we empirically verify the results using complicated learned optimizers with neural network parametrizations.

5.1.1 Our Results

In this paper we focus on two basic questions on learning-to-learn for gradient descent optimizer. First, will the meta-gradient explode/vanish and is there a way to fix the problem? Second, how could we guarantee that the learned optimizer has good generalization properties?

Our first result shows that meta-gradient can explode/vanish even for tuning the step size for gradient descent on a simple quadratic objective. In this setting, we show that there is a unique local and global minimizer for the step size, and we also give a simple way to get rid of the gradient explosion/vanishing problem.

Theorem 5.1.1 (Informal version of Theorem 5.3.1 and Theorem 5.3.2)**.** *For tuning the step size of gradient descent on a quadratic objective, if the meta-objective is the loss of the last iteration, then the meta-gradient will explode/vanish. If the meta-objective is the* log *of the loss of the last iteration, then the meta-gradient is polynomially bounded. Further, doing meta-gradient descent with a meta step size of* $1/\sqrt{k}$ *(where* k *is the number of meta-gradient steps) provably converges to the optimal step size for the inner-optimizer.*

Surprisingly, even though taking the log of the objective solves the meta-gradient explosion/vanishing problem, one cannot simply implement such an algorithm using back-propagation (which is standard in auto-differentiation tools such as those used

in TensorFlow (Abadi et al., 2016)). The reason is that even though the meta-gradient is polynomially bounded, back-propagation algorithm will compute the meta-gradient as the ratio of two exponentially large/small numbers, which causes numerical issues. Detailed discussion for the first result appears in Section 5.3.

Our second result shows that defining meta-objective on the same training set (later referred to as the "train-by-train" approach) could lead to overfitting; while defining meta-objective on a separate validation set ("train-by-validation", see Metz et al. (2019)) can solve this issue. We consider a simple least squares setting where $y = \langle w^* \rangle x + \xi$ and $\xi \sim \mathcal{N}(0, \sigma^2)$. We show that when the number of samples is small and the noise is large, it is important to use train-by-validation; while when the number of samples is much larger train-by-train can also learn a good optimizer.

Theorem 5.1.2 (Informal version of Theorem 5.4.1 and Theorem 5.4.2)**.** *For a least squares problem in d dimensions, if the number of samples n is a constant fraction of d (e.g., $d/2$), and the samples have large noise, then the train-by-train approach performs much worse than train-by-validation. On the other hand, when the number of samples n is large, train-by-train can get close to error $d\sigma^2/n$, which is optimal.*

We discuss the details in Section 5.4. In Section 5.5 we show that such observations also hold empirically for more complicated learned optimizers—an optimizer parametrized by a neural network.

5.1.2 Related Work

Learned optimizer The idea of learning an optimizer has appeared in early works decades ago (Bengio et al., 1990, 1992; Hochreiter et al., 2001). Recently, with the rise of deep learning, researchers started to consider more complex optimizers on more challenging tasks. One line of research views the optimizer as a policy and

apply reinforcement learning techniques to train it (Li and Malik, 2016, 2017; Bello et al., 2017). The other line of papers use gradient descent on the meta-objective to update the optimizer parameters (Maclaurin et al., 2015; Andrychowicz et al., 2016; Lv et al., 2017; Wichrowska et al., 2017; Metz et al., 2019).

Mostly relevant to our work, Metz et al. (2019) highlighted several challenges in the meta-optimization for learning-to-learn approach. First, they observed the meta-gradient exploding/vanishing issue and proposed to use a gradient estimator for a variational meta-objective. They also observed that train-by-train approach can overfit the training tasks while train-by-validation generalizes well.

Data-driven algorithm design In data-driven algorithm design, we aim to find an algorithm that works well on a particular distribution of tasks. Gupta and Roughgarden (2017) first modeled this algorithm-selection process as a statistical learning problem. In particular, they analyzed the sample complexity of choosing the step size for gradient descent. But they didn't consider the meta-optimization problem. They also restricted the step size into a small range so that gradient descent is guaranteed to converge on every task. We don't have such a restriction and allow the meta-learning to choose a more aggressive step size.

Following the work by Gupta and Roughgarden (2017), data-driven algorithms have been studied in many problems, including partitioning and clustering (Balcan et al., 2016a), tree search (Balcan et al., 2018b), pruning (Alabi et al., 2019) and machanism design (Morgenstern and Roughgarden, 2015, 2016; Balcan et al., 2016b, 2018a).

Step size schedule for GD/SGD Shamir and Zhang (2013) showed that SGD with polynomial step size scheduling can almost match the minimax rate in convex non-

smooth settings, which was later tightened by Harvey et al. (2018) for standard step size scheduling. Assuming that the number of training steps is known to the algorithm, the information-theoretically optimal bound in convex non-smooth setting was later achieved by Jain et al. (2019) which used another step size schedule, and Ge et al. (2019) showed that exponentially decaying step size scheduling can achieve near optimal rate for least squares regression.

A closely related paper that appeared later than our work also studied the comparison between train-by-train and train-by-validation (Bai et al., 2020). They considered a very different meta-learning problem, where the goal is to find the best common initialization for adapting to a linear predictor on each task. They proved train-by-train can work better than train-by-validation in the noiseless setting.

5.2 Preliminaries

In this section, we first introduce some notations, then formulate the learning-to-learn framework.

5.2.1 Notations

For any integer n, we use $[n]$ to denote $\{1, 2, \cdots, n\}$. We use $\|\cdot\|$ to denote the ℓ_2 norm for a vector and the spectral norm for a matrix. We use $\langle\cdot\rangle\cdot$ to denote the inner product of two vectors. For a symmetric matrix $A \in \mathbb{R}^{d\times d}$, we denote its eigenvalues as $\lambda_1(A) \geqslant \cdots \geqslant \lambda_d(A)$. We denote the d-dimensional identity matrix as I_d or simply as I when the dimension is clear. We use $O(\cdot), \Omega(\cdot), \Theta(\cdot)$ to hide constant factor dependencies. We use $\mathrm{poly}(\cdot)$ to represent a polynomial on the relevant parameters with constant degree.

5.2.2 Learning-to-Learn Framework

We consider the learning-to-learn approach applied to training a distribution of learning tasks. Each task is specified by a tuple $(\mathcal{D}, S_{\text{train}}, S_{\text{valid}}, \ell)$. Here $\mathcal{D}$ is a distribution of samples in $X \times Y$, where X is the domain for the sample and Y is the domain for the label/value. The sets S_{train} and S_{valid} are samples generated independently from $\mathcal{D}$, which serve as the training and validation set (the validation set is optional). The learning task looks to find a parameter $w \in W$ that minimizes the loss function $\ell(w, x, y) : W \times X \times Y \to \mathbb{R}$, which gives the loss of the parameter w for sample (x, y). The training loss for this task is

$$\widehat{f}(w) := \frac{1}{|S_{\text{train}}|} \sum_{(x,y) \in S_{\text{train}}} \ell(w, x, y),$$

while the population loss is $f(w) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(w, x, y)]$.

The goal of inner-optimization is to minimize the population loss $f(w)$. For the learned optimizer, we consider it as an update rule $u(\cdot)$ on weight w. The update rule is a parameterized function that maps the weight at step τ and its history to the step $\tau + 1 : w_{\tau+1} = u(w_\tau, \nabla\widehat{f}(w_\tau), \nabla\widehat{f}(w_{\tau-1}), \cdots; \theta)$. In most parts of this paper, we consider the update rule u as gradient descent mapping with step size as the trainable parameter (here $\theta = \eta$ which is the step size for gradient descent). That is, $u(w; \eta) = w - \eta\nabla\widehat{f}(w)$ for gradient descent and $u(w; \eta) = w - \eta\nabla_w\ell(w, x, y)$ for stochastic gradient descent where (x, y) is a sample randomly chosen from the training set S_{train}.

In the outer (meta) level, we consider a distribution $\mathcal{T}$ of tasks. For each task $P \sim \mathcal{T}$, we can define a meta-loss function $\Delta(\theta, P)$. The meta-loss function measures the performance of the optimizer on this learning task. The meta-objective, for

example, can be chosen as the target training loss $\widehat{f}$ at the last iteration (train-by-train), or the loss on the validation set (train-by-validation).

The training loss for the meta-level is the average of the meta-loss across m different specific tasks $P_1, P_2, ..., P_m$, that is,

$$\widehat{F}(\theta) = \frac{1}{m}\sum_{i=1}^{m}\Delta(\theta, P_k).$$

The population loss for the meta-level is the expectation over all the possible specific tasks $F(\theta) = \mathbb{E}_{P\sim\mathcal{T}}[\Delta(\theta, P)]$.

In order to train an optimizer by gradient descent, we need to compute the gradient of meta-objective $\widehat{F}$ in terms of meta parameters θ. The meta parameter is updated once after applying the optimizer on the inner objective t times to generate the trajectory $w_0, w_1, ..., w_t$. The meta-gradient is then computed by unrolling the optimization process and back-propagating through the t applications of the optimizer.

5.3 Alleviating Gradient Explosion/Vanishing Problems

First we consider the meta-gradient explosion/vanishing problem. More precisely, we say the meta-gradient explodes/vanishes if it is exponentially large/small with respect to the number of steps t of the inner-optimizer.

In this section, we consider a simple instance of the learning-to-learn approach, where the distribution $\mathcal{T}$ only contains a single task P, and the task also just defines a single loss function f[1]. Therefore, in this section $\widehat{F}(\eta) = F(\eta) = \Delta(\eta, P)$. We will simplify notation and only use $\widehat{F}(\eta)$.

[1] In the notation of Section 5.2, one can think that $\mathcal{D}$ contains a single point $(0, 0)$ and the loss function $f(w) = \ell(w, 0, 0)$.

The inner task P is a simple quadratic problem, where the starting point is fixed at w_0 with unit norm, and the loss function is $f(w) = \frac{1}{2} w^\top H w$ for some fixed positive definite matrix $H \in \mathbb{R}^{d \times d}$.

Let $\{w_{\tau,\eta}\}_{\tau=0}^{t}$ be the GD sequence running on $f(w)$ starting from w_0 with step size η. We consider two ways of defining meta-objective: using the loss of the last point directly or using the log of this value. We first show that although choosing $\widehat{F}(\eta) = f(w_{t,\eta})$ does not have any bad local optimal solution, it has the meta-gradient explosion/vanishing problem. We use $\widehat{F}'(\eta)$ to denote the derivative of $\widehat{F}$ in η.

In the analysis, we use eigen-decomposition to transform H into a diagonal matrix. We introduce related notations here: suppose the eigenvalue decomposition of H is $\sum_{i=1}^{d} \lambda_i u_i u_i^\top$. We denote $L := \lambda_1(H)$ and $\alpha := \lambda_d(H)$ as the largest and smallest eigenvalues of H. For each $i \in [d]$, let c_i be $\langle w_0 \rangle u_i$ and let $c_{\min}$ be $\min(|c_1|, |c_d|)$. We assume $c_{\min} > 0$ and $L > \alpha$ for simplicity[2].

Theorem 5.3.1. *Let the meta-objective be $\widehat{F}(\eta) = f(w_{t,\eta})$, we know $\widehat{F}(\eta)$ is a strictly convex function in η with an unique minimizer. However, for any step size $0 < \eta < 2/L$,*

$$|\widehat{F}'(\eta)| \leqslant tL^2 \max(|1 - \eta\alpha|^{2t-1}, |1 - \eta L|^{2t-1});$$

for any step size $\eta > 2/L$,

$$|\widehat{F}'(\eta)| \geqslant c_1^2 L^2 t (\eta L - 1)^{2t-1} - L^2 t.$$

Note that in Theorem 5.3.1, when $0 < \eta < 2/L$, $|\widehat{F}'(\eta)|$ is exponentially small because $|1 - \eta\alpha|, |1 - \eta L| < 1$; when $\eta > 2/L$, $|\widehat{F}'(\eta)|$ is exponentially large because $\eta L - 1 > 1$. The strict convexity of $\widehat{F}(\eta)$ is proved by showing the second order

[2] If w_0 is uniformly sampled from the unit sphere, with high probability $c_{\min}$ is at least $\Omega(1/\sqrt{d})$; if H is $XX^\top$ with $X \in \mathbb{R}^{d \times 2d}$ as a random Gaussian matrix, with constant probability, both α and $L - \alpha$ are at least $\Omega(d)$.

derivative of $\widehat{F}(\eta)$ is positive; the upper and lower bounds of $\widehat{F}'(\eta)$ follows from direct calculation.

Intuitively, gradient explosion/vanishing happens because the meta-objective becomes too small or too large. A natural idea to fix the problem is to take the log of the meta-objective to reduce its range. If we choose $\widehat{F}(\eta) = \frac{1}{t}\log f(w_{t,\eta})$, we have

Theorem 5.3.2. *Let the meta-objective be $\widehat{F}(\eta) = \frac{1}{t}\log f(w_{t,\eta})$. We know $\widehat{F}(\eta)$ has a unique minimizer η^* and $\widehat{F}'(\eta) = O\left(\frac{L^3}{c_{\min}^2\alpha(L-\alpha)}\right)$ for all $\eta \geqslant 0$. Let $\{\eta_k\}$ be the GD sequence running on $\widehat{F}$ with meta step size $\mu_k = 1/\sqrt{k}$. Suppose the starting step size $\eta_0 \leqslant M$. Given any $1/L > \epsilon > 0$, there exists $k' = \frac{M^6}{\epsilon^2}poly(\frac{1}{c_{\min}}, L, \frac{1}{\alpha}, \frac{1}{L-\alpha})$ such that for all $k \geqslant k'$, $|\eta_k - \eta^*| \leqslant \epsilon$.*

For convenience, in the above algorithmic result, we reset η to zero once η goes negative (this corresponds to doing a projected gradient descent on η under constraint $\eta \geqslant 0$). We give a proof sketch of Theorem 5.3.2 in Section 5.3.1.

Surprisingly, even though we showed that the meta-gradient is well-behaved, it cannot be effectively computed by doing back-propagation due to numerical issues. More precisely:

Corollary 5.3.3. *If we choose the meta-objective as $\widehat{F}(\eta) = \frac{1}{t}\log f(w_{t,\eta})$, when computing the meta-gradient using back-propagation, there are intermediate results that are exponentially large/small in number of inner-steps t.*

If we use back-propagation to compute $\widehat{F}'(\eta)$, we need to separately compute the numerator and denominator in Eqn. (5.1), which are exponentially large or small as we showed in Theorem 5.3.1. Indeed, in Section 5.5 we empirically verify that standard auto-differentiation tools can fail in this setting. In contrast, the meta training succeeds if we use the formula derived in Section 5.3.1 (Eqn. (5.2)). This

suggests that one should be more careful about using standard back-propagation in the learning-to-learn approach. The proofs of the results in this section are deferred into Appendix D.1.

5.3.1 Proof Sketch of Theorem 5.3.2

Throughout the proof, we work in the eigenspace of H which reduces the problem to having a diagonal matrix H. The proof goes in three steps:

- Claim 5.3.4 shows that the meta-objective $\widehat{F}$ has a unique minimizer η^* and the minus meta-gradient always points to the minimizer.
- Claim 5.3.5 shows meta-gradient $\widehat{F}'(\eta)$ never explodes.
- Claim 5.3.6 shows meta-gradient is large when η is far from η^*.

Claim 5.3.4. *The meta-objective $\widehat{F}$ has only one stationary point that is also its unique minimizer η^*. For any $\eta \in [0, \eta^*)$, $\widehat{F}'(\eta) < 0$ and for any $\eta \in (\eta^*, \infty)$, $\widehat{F}'(\eta) > 0$.*

The lemma follows from a direct calculation $\widehat{F}'(\eta)$:

$$\widehat{F}'(\eta) = \frac{-2\sum_{i=1}^{d} c_i^2 \lambda_i^2 (1-\eta\lambda_i)^{2t-1}}{\sum_{i=1}^{d} c_i^2 \lambda_i (1-\eta\lambda_i)^{2t}}. \tag{5.1}$$

Claim 5.3.4 is proved by noticing that the denominator in $\widehat{F}'(\eta)$ is always positive and the numerator is strictly increasing in η. Next, we show the meta derivative is polynomially upper bounded.

Claim 5.3.5. *For any $\eta \in [0, \infty)$, we have $|\widehat{F}'(\eta)| \leqslant \frac{4L^3}{c_{\min}^2 \alpha(L-\alpha)}$.*

To prove this claim we observe that the numerator and denominator are both polynomially bounded once we divide them by a common factor, which is $(1-\eta\alpha)^{2t}$

when $\eta \in [0, \frac{2}{\alpha+L}]$. More precisely we have when $\eta \in [0, \frac{2}{\alpha+L}]$

$$\left|\widehat{F}'(\eta)\right| = 2\frac{\left|\sum_{i=1}^{d} \frac{c_i^2\lambda_i^2}{1-\eta\alpha}\left(\frac{1-\eta\lambda_i}{1-\eta\alpha}\right)^{2t-1}\right|}{c_d^2\alpha + \sum_{i=1}^{d-1} c_i^2\lambda_i\left(\frac{1-\eta\lambda_i}{1-\eta\alpha}\right)^{2t}} \leqslant \frac{2\sum_{i=1}^{d} c_i^2\lambda_i^2}{c_d^2\alpha(1-\eta\alpha)}. \tag{5.2}$$

This leads to the claimed bounds based on our assumptions. The case when η is large is similar. Finally, we show the meta-gradient is lower bounded if η is away from η^* and is not too large. The proof follows from a similar calculation as above.

Claim 5.3.6. *Given $\widehat{M} \geqslant 2/\alpha$ and $1/L > \epsilon > 0$, for any $\eta \in [0, \eta^* - \epsilon] \cup [\eta^* + \epsilon, \widehat{M}]$, we have $|F'(\eta)| \geqslant 2\epsilon c_{\min}^2 \min\left(\frac{\alpha^3}{L}, \frac{1}{\widehat{M}^2}\right)$.*

With the above three claims, we are ready to sketch the proof of Theorem 5.3.2. Due to Claim 5.3.4, we know the minus meta-gradient always points to the minimizer η^*. This alone is not sufficient to prove the convergence result because the iterates might significantly overshoot the minimizer if $|\widehat{F}'|$ is too large or the iterates might converge very slowly if $|\widehat{F}'|$ is too small. Fortunately, these two problematic cases can be excluded by Claim 5.3.5 and Claim 5.3.6.

5.4 Generalization for Trained Optimizer

Next we consider the generalization ability of simple trained optimizers. In this section we consider a simple family of least squares problems. Let $\mathcal{T}$ be a distribution of tasks where every task $(\mathcal{D}(w^*), S_{\text{train}}, S_{\text{valid}}, \ell)$ is determined by a parameter $w^* \in \mathbb{R}^d$ that is sampled uniformly at random from the unit sphere. For each individual task, $(x, y) \sim \mathcal{D}(w^*)$ is generated by first choosing $x \sim \mathcal{N}(0, I_d)$ and then computing $y = \langle w^* \rangle x + \xi$ where $\xi \sim \mathcal{N}(0, \sigma^2)$ with $\sigma \geqslant 1$. The loss function $\ell(w, x, y)$ is just the squared loss $\ell(w, x, y) = \frac{1}{2}(y - \langle w \rangle x)^2$. That is, the tasks are just standard least-squares problems with ground-truth equal to w^* and noise level σ^2.

We consider two different ways to define the meta-objective.

Train-by-train: In the train-by-train setting, the training set S_{train} contains n independent samples, and the meta-loss function is chosen to be the training loss. That is, in each task P, we first choose w^* uniformly at random, then generate $(x_1, y_1), ..., (x_n, y_n)$ as the training set S_{train}. The meta-loss function $\Delta_{TbT(n)}(\eta, P)$ is defined to be

$$\Delta_{TbT(n)}(\eta, P) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \langle w_{t,\eta} \rangle x_i)^2.$$

Here $w_{t,\eta}$ is the result of running t iterations of gradient descent starting from point 0 with step size η. Note we truncate a sequence and declare the meta loss is high once the weight norm exceeds certain threshold[3]. We can safely do this because we assume the ground truth weight w^* has unit norm, so if the weight norm is too high, it means the inner training has diverged and the step size is too large.

As before, the empirical meta-objective in train-by-train setting is the average of the meta-loss across m different specific tasks $P_1, P_2, ..., P_m$, that is,

$$\widehat{F}_{TbT(n)}(\eta) = \frac{1}{m} \sum_{k=1}^{m} \Delta_{TbT(n)}(\eta, P_k). \tag{5.3}$$

Train-by-validation: In the train-by-validation setting, the specific tasks are generated by sampling n_1 training samples and n_2 validation samples for each task, and the meta-loss function is the validation loss. That is, in each specific task P, we first choose w^* uniformly at random, then generate $(x_1, y_1), ..., (x_{n_1}, y_{n_1})$ as the training set S_{train} and $(x'_1, y'_1), ..., (x'_{n_2}, y'_{n_2})$ as the validation set S_{valid}. The meta-loss function

[3] Specifically, if at the τ-th step $\|w_{\tau,\eta}\| \geqslant 40\sigma$, we freeze the training on this task and set $w_{\tau',\eta} = 40\sigma u$ for all $\tau \leqslant \tau' \leqslant t$, for some arbitrary vector u with unit norm. Setting the weight to a large vector is just one way to declare the loss is high.

$\Delta_{TbV(n_1,n_2)}(\eta, P)$ is defined to be

$$\Delta_{TbV(n_1,n_2)}(\eta, P) = \frac{1}{2n_2}\sum_{i=1}^{n_2}(y_i' - \langle w_{t,\eta}\rangle x_i')^2.$$

Here again $w_{t,\eta}$ is the result of running t iterations of the gradient descent on the training set starting from point 0, and we use the same truncation as before. The empirical meta-objective is defined as

$$\widehat{F}_{TbV(n_1,n_2)}(\eta) = \frac{1}{m}\sum_{k=1}^{m}\Delta_{TbV(n_1,n_2)}(\eta, P_k), \tag{5.4}$$

where each P_k is independently sampled according to the described procedure.

We first show that when the number of samples is small (in particular $n < d$) and the noise is a large enough constant, train-by-train can be much worse than train-by-validation, even when $n_1 + n_2 = n$ (the total number of samples used in train-by-validation is the same as in train-by-train)

Theorem 5.4.1. *Let $\widehat{F}_{TbT(n)}(\eta)$ and $\widehat{F}_{TbV(n_1,n_2)}(\eta)$ be as defined in Equation* (5.3) *and Equation* (5.4) *respectively. Assume $n, n_1, n_2 \in [d/4, 3d/4]$. Assume noise level σ is a large constant c_1. Assume unroll length $t \geqslant c_2$, number of training tasks $m \geqslant c_3 \log(mt)$ and dimension $d \geqslant c_4 \log(mt)$ for certain constants c_2, c_3, c_4. With probability at least* 0.99 *in the sampling of training tasks, we have*

$$\eta^*_{train} = \Theta(1) \text{ and } \mathbb{E}\left\|w_{t,\eta^*_{train}} - w^*\right\|^2 = \Omega(1)\sigma^2,$$

for all $\eta^*_{train} \in \arg\min_{\eta\geqslant 0}\widehat{F}_{TbT(n)}(\eta)$;

$$\eta^*_{valid} = \Theta(1/t) \text{ and } \mathbb{E}\left\|w_{t,\eta^*_{valid}} - w^*\right\|^2 = \|w^*\|^2 - \Omega(1)$$

for all $\eta^*_{valid} \in \arg\min_{\eta \geqslant 0} \widehat{F}_{TbV(n_1,n_2)}(\eta)$. *In both equations the expectation is taken over new tasks.*

In Theorem 5.4.1, $w_{t,\eta^*_{\text{train}}}$ and $w_{t,\eta^*_{\text{valid}}}$ are the results obtained on the new task and w^* is the ground truth of the new task. If σ is a large enough constant, we know $\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2$ is larger than $\mathbb{E}\left\|w_{t,\eta^*_{\text{valid}}} - w^*\right\|^2$ by some constant. The probability 0.99 is an arbitrary number, which can be replaced by any constant smaller than 1.

Note that in this case, the number of samples n is smaller than d, so the least square problem is under-determined and the optimal training loss would go to 0 (there is always a way to simultaneously satisfy all n equations). This is exactly what train-by-train would do—it will choose a large constant learning rate which guarantees the optimizer converges exponentially to the empirical risk minimizer (ERM)[4]. However, when the noise is large making the training loss go to 0 will overfit to the noise and hurt the generalization performance. In contrast, train-by-validation will choose a smaller learning rate which allows it to leverage the signal in the training samples without overfitting to noise.

We separately give a proof sketch for the train-by-train setting and train-by-validation setting in Section 5.4.1 and Section 5.4.2, respectively. The detailed proof of Theorem 5.4.1 is deferred to Appendix D.2. We also prove similar results for SGD in Appendix D.4

We emphasize that neural networks are often over-parameterized, which corresponds to the case when $d > n$. Therefore in order to train neural networks, it is usually better to use train-by-validation. On the other hand, we show when the number of samples is large ($n \gg d$), train-by-train can also perform well.

[4] In an under-determined problem, there are actually multiple ERM solutions. Here, we focus on the unique ERM solution in the span of training data. This is also the solution that GD converges to when the initialization is 0.

Theorem 5.4.2. *Let $\widehat{F}_{TbT(n)}(\eta)$ be as defined in Equation* (5.3)*. Assume noise level is a constant c_1. Given any $1 > \epsilon > 0$, assume training set size $n \geqslant \frac{cd}{\epsilon^2}\log\left(\frac{nm}{\epsilon d}\right)$, unroll length $t \geqslant c_2 \log\left(\frac{n}{\epsilon d}\right)$, number of training tasks $m \geqslant \frac{c_3 n^2}{\epsilon^4 d^2}\log\left(\frac{tnm}{\epsilon d}\right)$ and dimension $d \geqslant c_4$ for certain constants c, c_2, c_3, c_4. With probability at least* 0.99 *in the sampling of training tasks, we have*

$$\mathbb{E}\left\| w_{t,\eta^*_{train}} - w^* \right\|^2 \leqslant (1+\epsilon)\frac{d\sigma^2}{n},$$

for all $\eta^*_{train} \in \arg\min_{\eta \geqslant 0} \widehat{F}_{TbT(n)}(\eta)$, where the expectation is taken over new tasks.

Therefore if the learning-to-learn approach is applied to a traditional optimization problem that is not over-parameterized, train-by-train can work well. In this case, the empirical risk minimizer (ERM) already has good generalization performance, and train-by-train optimizes the convergence towards the ERM. We defer the proof of Theorem 5.4.2 into Appendix D.3.

5.4.1 Proof Sketch for Train-by-Train

In this section, we will give a proof sketch for the first half of Theorem 5.4.1 (train-by-train with small number of samples). At the end of this section, we will briefly discuss the proof of Theorem 5.4.2 (train-by-train with large number of samples). For convenience, we denote $\widehat{F}_{TbT}$ as the empirical meta-objective and F_{TbT} as the population meta-objective. We implicit assume the conditions in Theorem 5.4.1 hold in the following lemmas.

Our meta-optimization problem works on a distribution of tasks. Since different tasks can have different smoothness condition, it's possible that under the same step size, the inner training converges on some tasks, but diverges on others. One way to avoid this issue is to restrict the step size into a small range under which the inner

training converges on all tasks (Gupta and Roughgarden, 2017). But this is too conservative and may lead to suboptimal step size. Instead, we allow any positive step size and truncate the inner training if the weight norm goes too large. This approach resolves the diverging issues and also allow the meta-learning algorithm to choose a more aggressive step size. As we explain later, this brings some technical challenges into our proof.

In order to prove $\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2$ is large, we only need to show the population meta-objective $F_{TbT}(\eta^*_{\text{train}})$ is small. This is because $F_{TbT}(\eta^*_{\text{train}})$ measures the distance between $w_{t,\eta^*_{\text{train}}}$ and the ERM solution while ERM solution is far from w^*. Since η^*_{train} minimizes the empirical meta-objective, we know $\widehat{F}_{TbT}(\eta^*_{\text{train}})$ is small. Thus we only need to show F_{TbT} and $\widehat{F}_{TbT}$ are similar. This is easy to prove for small step sizes when the inner training always converges, but is difficult when the inner training can diverge and gets truncated. To address this problem we break the step size into three intervals separated by $1/L$ and $\widetilde{\eta}$ (L is a large constant that bounds the smoothness on all tasks). Intuitively, when $\eta \leqslant 1/L$ almost all inner training converges and larger step size leads to faster convergence and smaller $\widehat{F}_{TbT}$; on the other hand, when $\eta > \widetilde{\eta}$, we show $\widehat{F}_{TbT}(\eta)$ is always large so the minimizer of $\widehat{F}_{TbT}$ cannot be in this region. Therefore, the optimal step size must be in $[1/L, \widetilde{\eta}]$. We only need to prove in the interval $[1/L, \widetilde{\eta}]$ the empirical meta-objective $\widehat{F}_{TbT}$ is close to the population meta-objective F_{TbT}. This proof is still nontrivial since the inner training can still diverge on a small fraction of sampled tasks.

We first show that for $\eta \in [0, 1/L]$, the empirical meta-objective $\widehat{F}_{TbT}$ strictly decreases as η increases and $\widehat{F}_{TbT}$ is exponentially small in t at step size $1/L$.

Lemma 5.4.3. *With probability at least* $1 - m\exp(-\Omega(d))$, $\widehat{F}_{TbT}(\eta)$ *is monotonically*

decreasing in $[0, 1/L]$ *and*

$$\widehat{F}_{TbT}(1/L) \leqslant 2L^2\sigma^2\left(1 - \frac{1}{L^2}\right)^t.$$

Next we show that the minimizer cannot be larger than $\widetilde{\eta}$ for suitably chosen $\widetilde{\eta}$ (see the precise definition in the appendix). Intuitively, this is because when η is too large the inner-optimizer would diverge on a significant fraction of the sampled tasks.

Lemma 5.4.4. *With probability at least* $1 - \exp(-\Omega(m))$,

$$\widehat{F}_{TbT}(\eta) \geqslant \frac{\sigma^2}{10L^8}$$

for all $\eta > \widetilde{\eta}$.

By Lemma 5.4.3 and Lemma 5.4.4, we know when t is large enough, the optimal step size η^*_{train} must lie in $[1/L, \widetilde{\eta}]$. We can also show $1/L < \widetilde{\eta} < 3/L$, so η^*_{train} is a constant. To relate the empirical loss at η^*_{train} to the population loss, we prove the following uniform convergence result when $\eta \in [1/L, \widetilde{\eta}]$.

Lemma 5.4.5. *With probability at least* $1 - m\exp(-\Omega(d)) - O(t+m)\exp(-\Omega(m))$,

$$|F_{TbT}(\eta) - \widehat{F}_{TbT}(\eta)| \leqslant \frac{\sigma^2}{L^3},$$

for all $\eta \in [1/L, \widetilde{\eta}]$.

The proof of this Lemma involves constructing special ϵ-nets for F_{TbT} and $\widehat{F}_{TbT}$ and showing that for each fixed η, $|F_{TbT}(\eta) - \widehat{F}_{TbT}(\eta)|$ is small with high probability using concentration inequalities.

Combining the above lemmas, we know the population meta-objective F_{TbT} is small at η^*_{train}, which means $w_{t,\eta^*_{\text{train}}}$ is close to the ERM solution. Since the ERM solution overfits to the noise in the training samples, we know $\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|$ has to be large.

Train-by-train with large number of samples: The proof of Theorem 5.4.2 follows the same strategy as above. We prove that under the optimal step size η^*_{train}, $w_{t,\eta^*_{\text{train}}}$ converges to the ERM solution. But with more samples, the ERM solution w_{ERM} becomes closer to the ground truth w^*. More precisely, we can prove $\mathbb{E}\|w_{\text{ERM}} - w^*\|^2$ is roughly $\frac{d\sigma^2}{n}$, which leads to the bound in Theorem 5.4.2.

5.4.2 Proof Sketch for Train-by-Validation

In this section, we give a proof sketch for the second half of Theorem 5.4.1. We denote $\widehat{F}_{TbV}$ as the empirical meta-objective and F_{TbV} as the population meta-objective.

The overall proof strategy is similar as before: we will show the empirical meta-objective is high when the step size is beyond certain threshold, and only prove generalization result for step sizes below this threshold. Under the train-by-validation meta-objective, the optimal step size η^*_{valid} is in order $\Theta(1/t)$. So we will choose a smaller threshold step size to be $1/L$.

When $\eta < 1/L$, we show that the learned signal is linear in ηt while the fitted noise is quadratic in ηt. So there exists certain step size in the order $\Theta(1/t)$ such that our model can leverage the signal in the training set without overfitting the noise. More precisely, we prove the following lemma.

Lemma 5.4.6. *There exist $\eta_1, \eta_2, \eta_3 = \Theta(1/t)$ with $\eta_1 < \eta_2 < \eta_3$ such that*

$$F_{TbV}(\eta_2) \leqslant \frac{1}{2}\|w^*\|^2 - \frac{9}{10}C + \frac{\sigma^2}{2}$$

$$F_{TbV}(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{6}{10}C + \frac{\sigma^2}{2}, \forall \eta \in [0, \eta_1] \cup [\eta_3, 1/L]$$

where C is a positive constant.

We then show whenever η is large, either the gradient descent diverges and the sequence gets truncated or it converges and overfits the noise. In both cases, the meta-objective must be high.

Lemma 5.4.7. *With probability at least $1 - \exp(-\Omega(m))$,*

$$\widehat{F}_{TbV}(\eta) \geqslant C'\sigma^2 + \frac{1}{2}\sigma^2,$$

for all $\eta \geqslant 1/L$, where C' is a positive constant independent with σ.

To relate the behavior of F_{TbV} to the behavior of $\widehat{F}_{TbV}$, we prove the following uniform convergence result for step sizes in $[0, 1/L]$. The proof is similar as in Lemma 5.4.5.

Lemma 5.4.8. *With probability at least $1 - O(1/\epsilon)\exp(-\Omega(\epsilon^2 m))$,*

$$|\widehat{F}_{TbV}(\eta) - F_{TbV}(\eta)| \leqslant \epsilon,$$

for all $\eta \in [0, 1/L]$.

By choosing a small enough ϵ in Lemma 5.4.8, we ensure that the behavior of $\widehat{F}_{TbV}$ is similar as that of F_{TbV} in Lemma 5.4.6. Combing with Lemma 5.4.7, we know $\eta^*_{\text{valid}} = \Theta(1/t)$ and $F_{TbV}(\eta^*_{\text{valid}}) \leqslant \frac{1}{2}\|w^*\|^2 + \frac{1}{2}\sigma^2 - \Omega(1)$. This concludes our proof since $F_{TbV}(\eta) = \frac{1}{2}\mathbb{E}\|w_{t,\eta} - w^*\|^2 + \frac{1}{2}\sigma^2$.

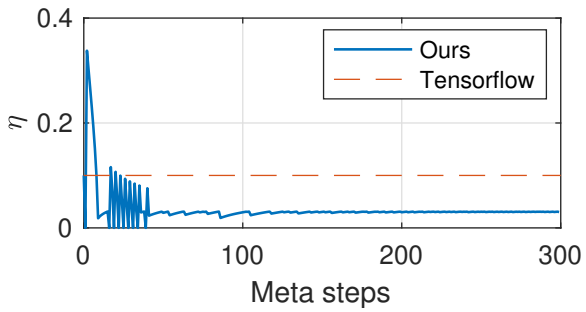


FIGURE 5.1: Meta training trajectory for η ($t = 80$, $\eta_0 = 0.1$).

5.5 Experiments

In this section, we give experiment results on both synthetic data and realistic data to verify our theory.[5]

Optimizing step size for quadratic objective We first validate the results in Section 5.3. We fixed a 20-dimensional quadratic objective as the inner problem and vary the number of inner steps t and initial value η_0. We compute the meta-gradient directly using the formula in Eqn. (5.2). In this way, we avoid the computation of exponentially small/large intermediate terms. We use the algorithm suggested in Theorem 5.3.2, except we choose the meta-step size to be $1/(100\sqrt{k})$ as the constants in the theorem were not optimized.

An example training curve of η for $t = 80$ and $\eta_0 = 0.1$ is shown in Figure 5.1, and we can see that η converges quickly within 300 steps. Similar convergence also holds for larger t or larger initial η_0. In contrast, we also implemented the meta-training with Tensorflow, where the code was adapted from the previous work of Wichrowska et al. (2017). Experiments show that in many settings (especially with large t and large η_0) the implementation does not converge. In Figure 5.1, under the TensorFlow implementation, the step size is stuck at the initial value throughout

[5] Our code is available at https://github.com/Kolin96/learning-to-learn.

the meta training because the meta-gradient explodes and gives NaN value. More details can be found in Appendix D.6.

Train-by-train vs. train-by-validation, synthetic data Here we validate our theoretical results in Section 5.4 using the least-squares model defined there. We fix the input dimension d to be 1000.

In the first experiment, we fix the size of the data ($n = 500$ for train-by-train, $n_1 = n_2 = 250$ for train-by-validation). Under different noise levels, we find the optimal η^* by a grid search on its meta-objective for train-by-train and train-by-validation settings respectively. We then use the optimal η^* found in each of these two settings to test on 10 new least-squares problem. The mean RMSE, as well as its range over the 10 test cases, are shown in Figure 5.2. We can see that for all of these cases, the train-by-train model overfits easily, while the train-by-validation model performs much better and does not overfit. Also, when the noise becomes larger, the difference between these two settings becomes more significant.

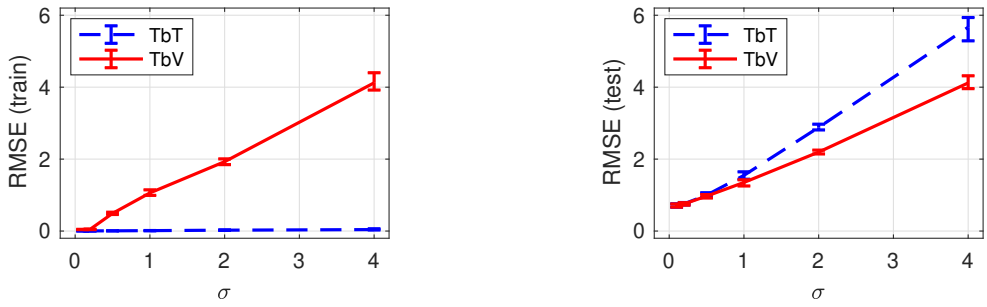


FIGURE 5.2: Training and testing RMSE for different σ values (500 samples)

In the next experiment, we fix $\sigma = 1$ and change the sample size. For train-by-validation, we always split the samples evenly into training and validation set. From Figure 5.3, we can see that the gap between these two settings is decreasing as we use more data, as expected by Theorem 5.4.2.

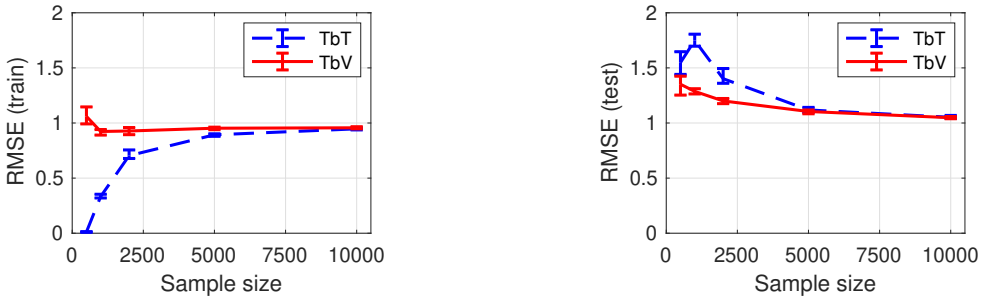


FIGURE 5.3: Training and testing RMSE for different samples sizes ($\sigma = 1$)

Train-by-train vs. train-by-validation, MLP optimizer on MNIST Here we consider the more interesting case of a multi-layer perceptron (MLP) optimizer on MNIST data set. We use the same MLP optimizer as in Metz et al. (2019), and details of this optimizer is discussed in Appendix D.6. As the inner problem, we use a two-layer fully-connected network of 100 and 20 hidden units with ReLU activations. The inner objective is the classic 10-class cross entropy loss, and we use mini-batches of 32 samples at inner training. In all the following experiments, we use SGD as a baseline with step size tuned by grid search against validation loss. For each optimizer, we run 5 independent tests and collect training accuracy and test accuracy for evaluation. The plots show the mean of the 5 tests[6].

In Figure 5.4, we show the test accuracy for different optimizers for different sample size and noise level. In this figure, "TbTx" represents train-by-train approach with x training samples; "TbV$x + y$" represents train-by-validation approach with x training samples and y validation samples. In Figure 5.4(a) the optimizer is applied to 1000 randomly sub-sampled data (split between training and validation for train-by-validation); in Figure 5.4(b) we use the same amount of data, except we add 20% label noise; in Figure 5.4(c) we use the whole MNIST dataset without label

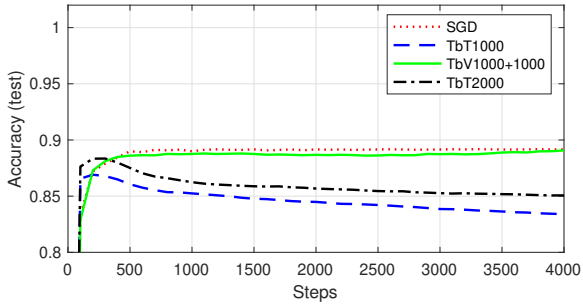
[6] We didn't show the measure of the spread because the results of these 5 tests are so close to each other, such that the range or standard deviation marks will not be readable in the plots.

noise. Comparing Figure 5.4(a) and (b), we see that when the noise is large train-by-validation significantly outperforms train-by-train. Figure 5.5 gives the training accuracy in the same setting as Figure 5.4(b), which clearly shows that train-by-validation can avoid overfitting to noisy labels. Comparing Figure 5.4(a) and (c), we see that when the number of samples is large enough there is no significant difference between train-by-train and train-by-validation.

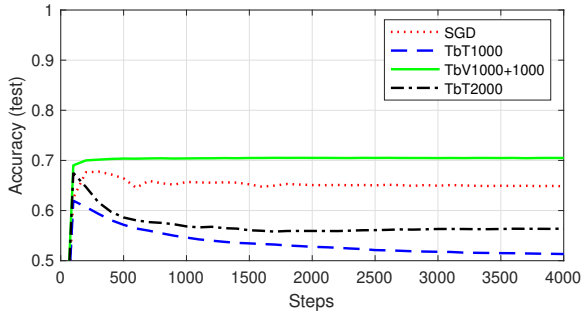
5.6 Conclusions

In this paper, we have proved optimization and generalization guarantees for tuning the step size for quadratic loss. From the optimization perspective, we considered a simple task whose objective is a quadratic function. We proved that the meta-gradient can explode/vanish if the meta-objective is simply the loss of the last iteration; we then showed that the log-transformed meta-objective has polynomially bounded meta-gradient and can be successfully optimized. To study the generalization issues, we considered the least squares problem—when the number of samples is small and the noise is large, train-by-validation approach generalizes better than train-by-train; while when the number of samples is large, train-by-train can also work well.

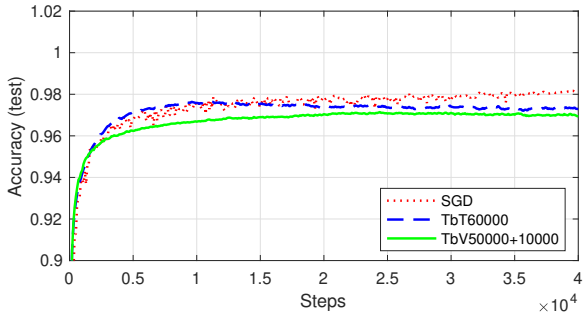
Although our theoretical results are proved for quadratic loss, this simple setting already yields interesting phenomenons and requires non-trivial techniques to analyze. We have also verified our theoretical results on an optimizer parameterized by neural networks and on MNIST dataset. There are still many open problems, including extending similar analysis to more complicated optimizers, or generalizing the idea to prevent numerical issues to neural network optimizers. We hope our work can lead to more theoretical understanding of the learning-to-learn approach.



(a) 1000 samples, no noise



(b) 1000 samples, 20% noise



(c) All samples, no noise

FIGURE 5.4: The test accuracy of different optimizers in various settings. Comparison between (a) and (b) shows that the advantage of train-by-validation over train-by-train increases when the samples have more noise; comparison between (a) and (c) shows that when the number of samples increases, train-by-train gets comparable performance as train-by-validation.

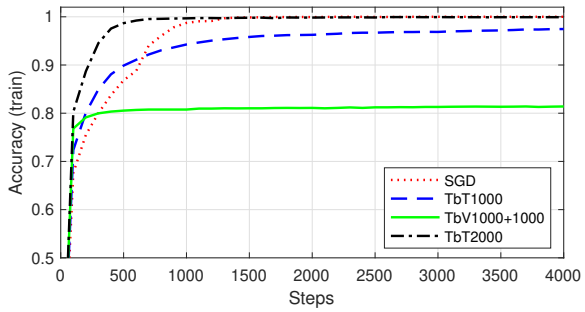


FIGURE 5.5: Training accuracy for 1000 samples and 20% noise (same setting as in Figure 5.4(b))

6

Conclusion

Self-supervised representation learning is a paradigm where representations are learned automatically during pre-training and later used in various downstream tasks. This thesis presents our explanation of the empirical success of self-supervised representation learning from two important perspectives. We analyze why the representations learned during pre-training can benefit downstream tasks. We also study the top eigenspaces of neural network Hessian and learning-to-learn for step size tuning to understand the optimization of neural networks.

Our analysis of the performance transfer from pre-training to downstream tasks is limited to simple or black-box models. A complete understanding of this requires analyzing more realistic white-box models. There are also many other important heuristics in neural network optimization that lack enough theoretical understanding. We believe our framework could potentially be applied there to provide more insights and inspire new algorithms.

Appendix A

Supplementary Materials for Chapter 2

A.1 Full Proofs

A.1.1 Proof of Theorem 2.3.2

We first recall the setting of Theorem 2.3.2.

Assumption 2.3.1. Let $A \in \mathbb{R}^{d\times p}$ be an arbitrary matrix with unit-norm columns satisfying $(2k, \delta)$-RIP for $k = o\left(\frac{d}{\log p}\right)$ and $\delta = o(1)$, and suppose $\sigma^2_{\min}(A) = \Omega(p/d)$. We assume each measurement y is generated as $y \sim Az + \epsilon$, where z is a random vector drawn from an arbitrary probability measure $\mathbb{P}_z$ on k-sparse vectors in $\mathbb{R}^p$, and $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ for some $\sigma > 0$.

And now we prove:

Theorem 2.3.2. *[Overfitting to Reconstruction Loss] Consider the data-generating model in Assumption 2.3.1 and define $\Lambda(z)$ to be:*

$$\Lambda(z) = \inf\{t \mid \mathbb{P}_z(z \geqslant t) \leqslant 1/d\}. \tag{2.8}$$

Then for $q = \Omega(p^2 \max(d\sigma^2, \Lambda(z)^2)/\sigma^2)$*, there exists a* $B \in R^{d \times q}$ *such that* $L(B,k) \leqslant L(A,k) - \Omega(k\sigma^2)$ *and* $d_R(A,B) = \Omega(\sigma^2)$.

Proof. Our proof technique will be to first lower bound the gap $L(A,k) - L(A,2k)$, and then to construct a B matrix that closely approximates the $2k$-sparse combinations of the columns of A.

From the definition of $L(B,k)$ we have that:

$$L(A,k) - L(A,2k) = \mathbb{E}_y \left[\min_{\|\widehat{z}\|_0 = k} \|y - A\widehat{z}\|^2 \right] - \mathbb{E}_y \left[\min_{\|\widehat{z}\|_0 = 2k} \|y - A\widehat{z}\|^2 \right] \tag{A.1}$$

Now let $\widehat{z}^*(y) = \arg\min_{\|\widehat{z}\|_0 = k} \|y - A\widehat{z}\|$ and $S^* = \operatorname{supp}(\widehat{z}^*)$, and further define:

$$\widetilde{z}(y) = \arg\min_{\|\widehat{z}\|_0 = k} \|(y - A\widehat{z}^*(y)) - A\widehat{z}\| \tag{A.2}$$

We will also use $\widetilde{S} = \operatorname{supp}(\widetilde{z}(y))$. For convenience, we will write $\widehat{z}^*$ and $\widetilde{z}$ when y is clear from context. Applying this notation to Equation (A.1) gives:

$$\begin{aligned} L(A,k) - L(A,2k) &\geqslant \mathbb{E}_y[\|y - A\widehat{z}^*\|^2] - \mathbb{E}_y[\|y - A\widehat{z}^* - A\widetilde{z}\|^2] \\ &= \mathbb{E}_y[2\langle y - A\widehat{z}^*, A\widetilde{z}\rangle] - \mathbb{E}_y[\|A\widetilde{z}\|^2] \\ &= \mathbb{E}_y[\|A\widetilde{z}\|^2] \end{aligned} \tag{A.3}$$

Where we obtained the last line above by using the fact that $A\widetilde{z}$ is the orthogonal projection of $y - A\widehat{z}^*$ on to the span of $A_{\widetilde{S}}$. Now using the fact that A is $(2k,\delta)$-RIP

we have that:

$$\begin{aligned}
\mathbb{E}_y[\|A\widetilde{z}\|^2] &\geqslant (1-\delta)^2\mathbb{E}_y\left[\|\widetilde{z}\|^2\right] \\
&\geqslant (1-\delta)^2\mathbb{E}_y\left[\left\|A_{\widetilde{S}}^{+}(y-A\widehat{z}^*)\right\|^2\right] \\
&\geqslant (1-\delta)^4\mathbb{E}_y\left[\left\|A_{\widetilde{S}}^{T}(y-A\widehat{z}^*)\right\|^2\right] \\
&= (1-o(1))\mathbb{E}_y\left[\left\|A_{\widetilde{S}}^{T}(y-A\widehat{z}^*)\right\|^2\right] \qquad \text{(A.4)}
\end{aligned}$$

Where above we used the fact that $A_{\widetilde{S}}^{+} = (A_{\widetilde{S}}^T A_{\widetilde{S}})^{-1}A_{\widetilde{S}}^T$ and RIP to obtain $\left\|(A_{\widetilde{S}}^T A_{\widetilde{S}})^{-1}\right\|_{op} \geqslant 1/(1+\delta)$, which led to the penultimate step. It remains to compute (or lower bound) the expectation in Equation (A.4). Towards this end, we let S denote a (uniformly) random subset of size k from $[p]$. Then we have that (using Assumption 2.3.1):

$$\begin{aligned}
\mathbb{E}_y\left[\left\|A_{\widetilde{S}}^{T}(y-A\widehat{z}^*)\right\|^2\right] &\geqslant \mathbb{E}_y\left[\mathbb{E}_S\left[\left\|A_{S}^{T}(y-A\widehat{z}^*)\right\|^2\right]\right] \\
&= \frac{k}{p}\mathbb{E}_y\left[\left\|A^{T}(y-A\widehat{z}^*)\right\|^2\right] \\
&\geqslant \frac{k}{p}\sigma_{\min}^2(A)\mathbb{E}_y\left[\|y-A\widehat{z}^*\|^2\right] \\
&= \Omega\left(\frac{k}{d}\mathbb{E}_y\left[\|y-A\widehat{z}^*\|^2\right]\right) \qquad \text{(A.5)}
\end{aligned}$$

Now the expectation in Equation (A.5) can be lower bounded in the same vein as Equation (A.4) (i.e. relying on the RIP property). Below we use S_{2k}^* to denote

the optimal support for the minimization problem.

$$\begin{aligned}
\mathbb{E}_y\left[\|y - A\widehat{z}^*\|^2\right] &= \mathbb{E}_{z,\epsilon}\left[\min_{\|\widehat{z}\|_0=k} \|\epsilon - A(\widehat{z} - z)\|^2\right] \\
&\geqslant \mathbb{E}_{\epsilon}\left[\min_{\|\widehat{z}\|_0=2k} \|\epsilon - A\widehat{z}\|^2\right] \\
&= \mathbb{E}_{\epsilon}\left[\|\epsilon\|^2\right] - 2\mathbb{E}_{\epsilon}\left[\left\langle A^+_{S^*_{2k}}\epsilon, \epsilon\right\rangle\right] + \mathbb{E}_{\epsilon}\left[\left\|A^+_{S^*_{2k}}\epsilon\right\|^2\right] \\
&\geqslant \mathbb{E}_{\epsilon}\left[\|\epsilon\|^2\right] - (1 + o(1))\mathbb{E}_{\epsilon}\left[\left\|A^T_{S^*_{2k}}\epsilon\right\|^2\right] \\
&\geqslant d\sigma^2 - (1 + o(1))2k\mathbb{E}_{\epsilon}\left[\max_{i\in[p]}(A_i^T\epsilon)^2\right] \\
&\geqslant d\sigma^2 - O\left(k\sigma^2 \log p\right) \\
&= \Omega(d\sigma^2)
\end{aligned} \tag{A.6}$$

Where above we used Lemma A.1.2 since the random variables $(A_i^T\epsilon)^2$ follows a scaled chi-square distribution with degree of freedom 1, and for the last line we use $k = o\left(\frac{d}{\log p}\right)$. Now putting Equations (A.3)-(A.6) together, we obtain:

$$L(A, k) - L(A, 2k) \geqslant \Omega(k\sigma^2) \tag{A.7}$$

Given the gap between $L(A, k)$ and $L(A, 2k)$ shown in Equation (A.7), our goal is now to construct a matrix B such that we can approximate sufficiently large $2k$-sparse combinations of the columns of A via $B\widehat{z}$ (where $\widehat{z}$ is k-sparse). We recall from standard concentration of measure arguments (see Vershynin (2018a) for details) that $\mathbb{P}(\|\epsilon\|^2 \geqslant 2d\sigma^2) \leqslant \exp(-\Omega(d))$. Furthermore, by Assumption 2.3.1, $\|Az\| \leqslant \Lambda(z)(1 + o(1))$ with probability $1 - 1/d$. Thus, we only need the columns of B to approximate Ax for 2-sparse x (since we are interested in $B\widehat{z}$ and $\widehat{z}$ is k-sparse) and $\|Ax\| \leqslant \gamma_1 \max(\sigma\sqrt{d}, \Lambda(z))$ for an appropriately large constant γ_1 (as this will imply

we get the same gap as Equation A.6).

To do this, we can construct ϵ-nets for each of the following sets (indexed by the different possible 2-sparse supports $S \subset [p]$):

$$V_S = \{Az \mid \text{supp}\,(z) = S,\ \|Az\| \leqslant \gamma_1 \max(\sigma\sqrt{d}, \Lambda(z))\} \tag{A.8}$$

Since A has p columns, we need $\Theta(p^2)$ such ϵ-nets. As long as we choose $\epsilon = \gamma_2\sigma^2$ with γ_2 a constant, we can approximate $2k$-sparse combinations of the columns of A with error $k\gamma_2\sigma^2$ using k-sparse combinations from these nets, which is sufficient for our purposes given the result of Equation (A.7).

Now let the columns of B be the union of the ϵ-nets for the sets V_S and define $\mathcal{E} = \{\|y\| \leqslant \gamma_1 \max(\sigma\sqrt{d}, \Lambda(z))\}$. After choosing γ_2 to be sufficiently small, we then get from Equations (A.3)-(A.7):

$$\begin{aligned} L(A,k) - L(B,k) &\geqslant \mathbb{E}_y\left[\|A\widetilde{z}\|^2 \mid \mathcal{E}\right]\ \mathbb{P}\left(\mathcal{E}\right) - k\gamma_2\sigma^2 \\ &= \Omega(k\sigma^2) \end{aligned} \tag{A.9}$$

Noting that the ϵ-nets for each V_S are of size $O(\max(d\sigma^2, \Lambda(z)^2)/\sigma^2)$ from our choice of ϵ (once again, refer to Vershynin (2018a) for bounds on the size of ϵ-nets), this construction of B requires $O(p^2 \max(d\sigma^2, \Lambda(z)^2)/\sigma^2)$ columns. As we can choose these columns to be different from those of A by $\gamma_2\sigma^2$ (in norm), we obtain the desired result. □

A.1.2 Proof of Theorem 2.3.6

Again, we first recall the setting of Theorem 2.3.6.

Assumption 2.3.5. Let $A \in \mathbb{R}^{d\times p}$ be an arbitrary matrix such that there exists an $M \subset [d]$ with $P_M A$ being μ-incoherent with $\mu \leqslant C/(2k-1)$ for a universal constant $C < 1$. We assume each measurement y is generated as $y \sim Az + \epsilon$, where

$[z]_{\text{supp}(z)} \sim \mathcal{N}(0, \sigma_z^2 \mathrm{I}_k)$ with $\text{supp}(z)$ drawn from an arbitrary probability distribution over all size-k subsets of $[d]$, and $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ for some $\sigma > 0$.

In order to prove Theorem 2.3.6, we will need a result from Cai and Wang (2011), which we restate below.

Theorem A.1.1 (Theorem 9 in Cai and Wang (2011))**.** *For $y \sim Az + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I})$ and $A \in \mathbb{R}^{d \times p}$ being μ-incoherent with $\mu < 1/(2k-1)$, let us define:*

$$\mathcal{S} = \left\{ A_i : i \in [p],\ |z_i| \geqslant \frac{2\sigma\sqrt{k}\sqrt{2(1+\eta)\log p}}{1-(2k-1)\mu} \right\} \tag{A.10}$$

Then OMP (as defined in Algorithm) selects a column from $\mathcal{S}$ at each step with probability at least $1 - \frac{1}{p^\eta \sqrt{2\log p}}$.

Now we may prove:

Theorem 2.3.6. *[Benefits of Masking] Consider the data-generating model in Assumption 2.3.5. For any non-empty mask $M \subset [d]$ such that $P_M A$ satisfies the μ-incoherence condition in the assumption, we have*

$$\lim_{\sigma_z \to \infty} \left(L_{mask}(A, k, M) - \min_B L_{mask}(B, k, M) \right) = 0 \tag{2.11}$$

That is, as the expected norm of the signal Az increases, there exist minimizers B of L_{mask} such that $d_R(A, B) \to 0$.

Proof. We have from the definition of L_{mask} that:

$$\begin{aligned} L_{mask}(B,k,M) &= \mathbb{E}_{z,\epsilon}\left[\left\|[Az]_{[d]\setminus M} + [\epsilon]_{[d]\setminus M} - [B\widehat{z}]_{[d]\setminus M}\right\|^2\right] \\ &= \mathbb{E}_{z,\epsilon}\left[\left\|[Az]_{[d]\setminus M} - [B\widehat{z}]_{[d]\setminus M}\right\|^2\right] + E_{\epsilon}\left[\left\|[\epsilon]_{[d]\setminus M}\right\|^2\right] \\ &\quad - \mathbb{E}_{z,\epsilon}\left[\left\langle [B\widehat{z}]_{[d]\setminus M}, [\epsilon]_{[d]\setminus M}\right\rangle\right] \\ &= \mathbb{E}_{z,\epsilon}\left[\left\|[Az]_{[d]\setminus M} - [B\widehat{z}]_{[d]\setminus M}\right\|^2\right] + (d-|M|)\sigma^2 \\ &= \mathbb{E}_{z,\epsilon}\left[\left\|P_{[d]\setminus M}Az - P_{[d]\setminus M}B\widehat{z}\right\|^2\right] + (d-|M|)\sigma^2 \end{aligned} \tag{A.11}$$

Since $[B\widehat{z}]_{[d]\setminus M}$ is necessarily independent[1] of $[\epsilon]_{d\setminus M}$ (by the construction of $\widehat{z}$). Now the quantity in Equation A.11 depending on B looks almost identical to the prediction risk considered in linear regression.

With this in mind, let us define:

$$\mathcal{R}_M(\widehat{y}) = \mathbb{E}_{z,\epsilon}\left[\left\|P_{[d]\setminus M}Az - \widehat{y}\right\|^2\right] \tag{A.12}$$

Where $\widehat{y}$ is any estimator that depends only on $[y]_M$ (i.e. in the interest of brevity we are omitting writing $\widehat{y}([Az+\epsilon]_M)$). We can lower bound Equation (A.11) by analyzing $\mathcal{R}_M(\widehat{y})$:

$$\begin{aligned} \inf_{\widehat{y}} \mathcal{R}_M &= \inf_{\widehat{y}} \mathbb{E}_{z,\epsilon}\left[\left\|P_{[d]\setminus M}Az - \widehat{y}\right\|^2\right] \\ &= \inf_{\widehat{y}} \mathbb{E}_{z,\epsilon}\left[\mathbb{E}_{z,\epsilon}\left[\left\|P_{[d]\setminus M}A_{\mathrm{supp}(z)}[z]_{\mathrm{supp}(z)} - \widehat{y}\right\|^2\right] \;\middle|\; \mathrm{supp}(z)\right] \end{aligned} \tag{A.13}$$

Equation (A.13) can be lower bounded by considering the infimum over the inner expectation with respect to estimators $\widehat{y}$ that have access to $[Az+\epsilon]_M$ and the

[1] There is actually a slight technicality here; we need g_{OMP} to be Borel measurable, which is the case because it consists of the composition of Borel measurable functions.

support $S^* = \text{supp}(z)$. In this case, the Bayes estimator $\widehat{y}^*$ is:

$$\begin{aligned}\widehat{y}^* &= \mathbb{E}_{z,\epsilon}\left[P_{[d]\setminus M}A_{S^*)}[z]_{S^*} \;\middle|\; P_M A_{S^*}[z]_{S^*} + [\epsilon]_M\right] \\ &= P_{[d]\setminus M}A_{S^*}\mathbb{E}_{z,\epsilon}\left[[z]_{S^*} \;\middle|\; P_M A_{S^*}[z]_{S^*} + [\epsilon]_M\right] \qquad \text{(A.14)}\end{aligned}$$

Since $[z]_{S^*} \sim \mathcal{N}(0, \sigma_z^2 \mathrm{I}_k)$, we can explicitly compute the conditional expectation in Equation (A.14). Indeed, it is just the ridge regression estimator:

$$\widehat{y}^* = P_{[d]\setminus M}A_{S^*}\left(\Lambda_{S^*}^T\Lambda_{S^*} + \frac{1}{\sigma_z^2}\mathrm{I}_k\right)^{-1}\Lambda_{S^*}^T(P_M A_{S^*}[z]_{S^*} + [\epsilon]_M) \qquad \text{(A.15)}$$

Where we have set $\Lambda_{S^*} = P_M A_{S^*}$ above to keep notation manageable. Thus, putting all of the above together we have:

$$L_{mask}(B, k, M) \geqslant \mathcal{R}_M(\widehat{y}^*) + (d - |M|)\sigma^2 \qquad \text{(A.16)}$$

Now let $\widehat{y}_{LS}$ be the least squares estimator with access to the support $\text{supp}(z)$:

$$\widehat{y}_{LS} = P_{[d]\setminus M}A_{S^*}\Lambda_{S^*}^+(P_M A_{S^*}[z]_{S^*} + [\epsilon]_M) \qquad \text{(A.17)}$$

Then we have $\mathcal{R}_M(\widehat{y}_{LS}) \to \mathcal{R}_M(\widehat{y}^*)$ as $\sigma_z^2 \to \infty$. If we show that $\mathcal{R}_M(P_{[d]\setminus M}A\widehat{z}) \to \mathcal{R}_M(\widehat{y}_{LS})$, then we will be done by Equation (A.16).

Showing this essentially boils down to controlling the error of $\widehat{z}$ when OMP fails to recover the true support S^* (because when it recovers the true support, $P_{[d]\setminus M}A\widehat{z}$ is exactly $\widehat{y}_{LS}$). We do this by appealing to Theorem A.1.1.

Recall that $[\widehat{z}]_S = \Lambda_S^+[y]_M$, where $\Lambda_S = P_M A_S$ with $S = \text{supp}(\widehat{z})$ being the support predicted by OMP. Letting $\widehat{z}_{LS}$ be the vector whose non-zero components correspond to $[\widehat{z}_{LS}]_{S^*} = \Lambda_{S^*}^+[y]_M$, it will suffice to show $\|\widehat{z}_{LS} - \widehat{z}\|^2 \to 0$ as $\sigma_z \to \infty$, since then we will be done due to the fact that $\left\|P_{[d]\setminus M}A\right\|_{op}$ is constant with respect to σ_z.

Now letting $\widetilde{z} = \widehat{z} - \Lambda_S^+[\epsilon]_M$ (i.e. $\widetilde{z}$ represents the part of the signal z recovered by OMP), we have:

$$\begin{aligned}\|\widehat{z}_{LS} - \widehat{z}\|^2 &= \left\|z - \widetilde{z} + (\Lambda_{S^*}^+ - \Lambda_S^+)[\epsilon]_M\right\|^2 \\ &\leqslant \|z - \widetilde{z}\|^2 + \left\|(\Lambda_{S^*}^+ - \Lambda_S^+)[\epsilon]_M\right\|^2 + 2\|z - \widetilde{z}\|\left\|(\Lambda_{S^*}^+ - \Lambda_S^+)[\epsilon]_M\right\| \quad \text{(A.18)}\end{aligned}$$

We begin by first analyzing $\|z - \widetilde{z}\|^2$. To do so, we introduce the notation $[z]_{U,0}$ to represent the vector in $\mathbb{R}^k$ whose non-zero entries correspond to $[z]_U$ for $U \subset [d],\ |U| \leqslant k$. Then we can make use of the following decomposition of $\Lambda_{S^*}[z]_{S^*}$:

$$\Lambda_{S^*}[z]_{S^*} = \Lambda_S[z]_{S^* \cap S,0} + \Lambda_{S^*}[z]_{S^* \setminus S,0} \quad \text{(A.19)}$$

From Equation (A.19) we get:

$$\begin{aligned}\|z - \widetilde{z}\|^2 &= \left\|[z]_{S^* \setminus S,0} - \Lambda_S^+ \Lambda_{S^*}[z]_{S^* \setminus S,0}\right\|^2 \\ &\leqslant \left\|[z]_{S^* \setminus S,0}\right\|^2 + \left\|(\Lambda_S^T \Lambda_S)^{-1} \Lambda_S^T \Lambda_{S^*}[z]_{S^* \setminus S,0}\right\|^2 \\ &\quad + 2\max\left(\left\|[z]_{S^* \setminus S,0}\right\|^2, \left\|\Lambda_S^+ \Lambda_{S^*}[z]_{S^* \setminus S,0}\right\|^2\right) \\ &= \sum_{i \in S^* \setminus S} O(z_i^2) \quad \text{(A.20)}\end{aligned}$$

where we passed from the penultimate to the last line by using the μ-incoherence of $P_M A$ to control the middle term in the bound. With Equation (A.20) in hand, we are finally in a position to apply Theorem A.1.1. Let $\eta = C' \log \sigma_z$ for a sufficiently large constant C'. Now for convenience we define:

$$\gamma = \frac{2\sigma\sqrt{k}\sqrt{2(1+\eta)\log p}}{1 - (2k-1)\mu} \quad \text{(A.21)}$$

which corresponds to the lower bound in Equation (A.10). Using Theorem A.1.1

with Equation (A.20) we obtain:

$$\mathbb{E}_{z,\epsilon}\left[\|z-\tilde{z}\|^2\right] \leqslant \sum_{i\in S^*} \mathbb{P}(\{|z_i| \geqslant \gamma\} \cap \{i \notin S\}) O(\mathbb{E}[z_i^2]) + \mathbb{P}(|z_i| < \gamma) O(\gamma^2)$$

$$= \sum_{i\in S^*} O\left(\frac{\sigma_z^2}{\sigma_z^{C'}\sqrt{\log p}} + \frac{\gamma^3}{\sigma_z}\right) \tag{A.22}$$

And clearly Equation (A.22) goes to 0 as $\sigma_z \to \infty$. We can apply similar analysis techniques to the term $\left\|(\Lambda^+_{S^*} - \Lambda^+_S)[\epsilon]_M\right\|^2$ in Equation (A.18) as well, but for this term we can afford to be less precise.

Namely, when $S = S^*$, this term is 0. The probability that $S \neq S^*$ can be bounded as:

$$\mathbb{P}(S \neq S^*) \leqslant O\left(\frac{k}{\sigma_z^{C'}\sqrt{\log p}}\right) \mathbb{P}(\min_{i\in S^*} |z_i| \geqslant \gamma)$$

$$= O\left(\frac{k\gamma^k}{\sigma_z^{C'+k}\sqrt{\log p}}\right) \tag{A.23}$$

where again above we used the naive bound for $\mathbb{P}(\min_{i\in S^*} |z_i| \geqslant \gamma)$ (i.e. replacing the density with 1 and integrating from $\pm\gamma$). Now we have:

$$\begin{aligned}
\mathbb{E}_{z,\epsilon}\left[\left\|(\Lambda^+_{S^*} - \Lambda^+_S)[\epsilon]_M\right\|^2\right] &\leqslant \mathbb{P}(S \neq S^*)\mathbb{E}_{z,\epsilon}\Big[\left\|\Lambda^+_{S^*}[\epsilon]_M\right\|^2 + \left\|\Lambda^+_S[\epsilon]_M\right\|^2 \\
&\quad + 2\max\left(\left\|\Lambda^+_{S^*}[\epsilon]_M\right\|^2, \left\|\Lambda^+_S[\epsilon]_M\right\|^2\right)\Big] \\
&\leqslant \mathbb{P}(S \neq S^*) O\left(k\mathbb{E}_\epsilon\Big[\max_{i\in[p]}(A_i^T\epsilon)^2\Big]\right) \\
&\leqslant \mathbb{P}(S \neq S^*) O(k\sigma^2 \log p)
\end{aligned} \tag{A.24}$$

Putting together Equations (A.22) and (A.24) shows that Equation (A.18) goes to 0 as $\sigma_z \to \infty$, which proves the result.

□

A.1.3 Auxiliary Lemmas

Lemma A.1.2. *Let $X_1, \cdots, X_n$ be n chi-square random variables with 1 degree of freedom, then*

$$\mathbb{E}\left[\max_{i\in[n]} X_i\right] = O(\log n).$$

Proof. We bound the maximum via the moment-generating function.

From Jensen's inequality, for $t \in (0, \frac{1}{2})$, we have

$$\begin{aligned}\exp\left(t \cdot \mathbb{E}\left[\max_{i\in[n]} X_i\right]\right) &\leqslant \mathbb{E}\left[\exp\left(t \cdot \max_{i\in[n]} X_i\right)\right] = \max_{i\in[n]} \mathbb{E}\left[e^{tX_i}\right] \\ &\leqslant \sum_{i=1}^{n} \mathbb{E}\left[e^{tX_i}\right] = n(1-2t)^{-\frac{1}{2}}.\end{aligned}$$

Setting $t = \frac{1}{3}$ gives us

$$\mathbb{E}\left[\max_{i\in[n]} X_i\right] \leqslant 3\log n - \frac{3}{2}\log\frac{1}{3} = O(\log n).$$

□

Appendix B

Supplementary Materials for Chapter 3

B.1 Proof of Theorem 3.3.1

Theorem 3.3.1. *Suppose the downstream task performance depends only on a function* $f^*(x,i) = \langle v^*_{-i}(x_{-i}), u^* \rangle = \sum_{t=1}^k c^*_t \langle v^*_{-i}(x_{-i}), v^*_t \rangle$. *For* $t^- \in [k]$, *define* $p^- := p^*(x_i = t^- | x_{-i})$, *and assume* $p^- \leqslant \frac{1}{2}$. *Then for all* $s \in \mathbb{R}^+$, *there exist functions* v_{-i} *and* $\{v_t\}_{t=1}^k$ *such that* $D_{\mathrm{KL}}(p^*(x_i|x_{-i}) || p(x_i|x_{-i})) \leqslant 2sp^-$ *and* $f(x,i) := \sum_{t=1}^k c^*_t \langle v_{-i}(x_{-i}), v_t \rangle \leqslant f^*(x,i) - s \cdot c^*_{t^-}$.

Proof. We choose v_{-i} and $\{v_{k_t}\}_{t=1}^r$ such that $\forall x, i,\ \forall t \in [n] \backslash \{t^-\}$, $\langle v_{-i}(x_{-i}), v_t \rangle = \langle v^*_{-i}(x_{-i}), v^*_t \rangle$. Besides, $\forall x, i$, $\langle v_{-i}(x_{-i}), v_{t^-} \rangle = \langle v^*_{-i}(x_{-i}), v^*_{t^-} \rangle - s$.

From the definition of p^- and t^- we know that

$$p^- = p^*(x_i = t^- | x_{-i}) = \frac{\exp\big(\langle v^*_{-i}, v^*_{t^-} \rangle\big)}{Z^*(x,i)}. \tag{B.1}$$

Following our construction of the student model, its partition function satisfies

$$Z(x,i) = \sum_{j=1}^{n} \exp(\langle v_{-i}, v_j \rangle) \tag{B.2}$$

$$> \sum_{j \neq t^-} \exp\big(\langle v^*_{-i}, v^*_j \rangle\big) = Z^*(x,i) - \exp\big(\langle v^*_{-i}, v^*_{t^-} \rangle\big) = (1-p^-)Z^*(x,i). \tag{B.3}$$

Besides,

$$Z(x,i) = \sum_{j=1}^{n} \exp(\langle v_{-i}, v_j \rangle) < \sum_{j=1}^{n} \exp\big(\langle v^*_{-i}, v^*_j \rangle\big) = Z^*(x,i). \tag{B.4}$$

Therefore,

$$\forall j \neq t^-, \quad \frac{p^*(x_i = j | x_{-i})}{p(x_i = j | x_{-i})} = \frac{\frac{\exp\left(\langle v^*_{-i}, v^*_j \rangle\right)}{Z^*(x,i)}}{\frac{\exp\left(\langle v^*_{-i}, v^*_j \rangle\right)}{Z(x,i)}} = \frac{Z(x,i)}{Z^*(x,i)}. \tag{B.5}$$

$$\frac{p^*(x_i = t^- | x_{-i})}{p(x_i = t^- | x_{-i})} = \frac{\frac{\exp\left(\langle v^*_{-i}, v^*_j \rangle\right)}{Z^*(x,i)}}{\frac{\exp\left(\langle v^*_{-i}, v^*_j \rangle - r\right)}{Z(x,i)}} = \frac{Z(x,i)}{Z^*(x,i)} \cdot e^s. \tag{B.6}$$

Thus,

$$D_{\mathrm{KL}}(p^*(x_i|x_{-i}) || p(x_i|x_{-i})) = \sum_{j=1}^{n} p^*(x_i = j | x_{-i}) \log \frac{p^*(x_i = j | x_{-i})}{p(x_i = j | x_{-i})} \tag{B.7}$$

$$= \sum_{j \neq t^-} p^*(x_i = j | x_{-i}) \log \frac{Z(x,i)}{Z^*(x,i)} + p^- \cdot s \log \frac{Z(x,i)}{Z^*(x,i)} \tag{B.8}$$

$$< s p^- \log \frac{1}{1-p^-} < 2sp^- \tag{B.9}$$

and

$$f(x,i) = \sum_{t=1}^{k} c^*_t \langle v_{-i}(x_{-i}), v_t \rangle = \sum_{t=1}^{k} c^*_t \langle v^*_{-i}(x_{-i}), v^*_t \rangle - c^*_t \cdot s = f^*(x,i) - s \cdot c^*_{t^-}. \tag{B.10}$$

□

B.2 Detailed Proofs of Auxilliary Lemmas

Lemma 3.5.10. $S(x) \neq \varnothing$.

Proof. Assume by way of contradiction that $S(x) = \varnothing$. Then by the definition of $S(x)$, we know that for all j such that $a_j^* > 0$,

$$\begin{aligned}&\sigma(\langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^*) - \sigma(\langle v_{-i}(x), v_j - v_0\rangle - b_j)\\ \leqslant&(\langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^*) - \sigma(\langle v_{-i}(x), v_j - v_0\rangle - b_j) = \Delta_j \leqslant 0.\end{aligned}$$

Therefore,

$$\sum_{j:a_j^*>0} a_j^* \left(\sigma(\langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^*) - \sigma(\langle v_{-i}(x), v_j - v_0\rangle - b_j)\right) \leqslant 0 < \frac{\gamma}{2},$$

which is a contradiction. Thus, $S(x) \neq \varnothing$. □

Lemma 3.5.11. $\sum_{j\in S(x)} \Delta_j \geqslant \frac{\gamma}{2\max_{j\in S(x)} a_j^*}$.

Proof. Note that for all $j \in S(x)$, $\langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^* > \sigma(\langle v_{-i}(x), v_j - v_0\rangle - b_j^*) \geqslant 0$, so $\sigma(\langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^*) = \langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^*$. Therefore,

$$\begin{aligned}\sum_{j\in S(x)} a_j^*\Delta_j &= \sum_{j\in S(x)} a_j^* \left((\langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^*) - \sigma((\langle v_{-i}(x), v_j - v_0\rangle - b_j^*))\right) &\text{(B.11)}\\ &= \sum_{j\in S(x)} a_j^* \left(\sigma(\langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^*) - \sigma(\langle v_{-i}(x), v_j - v_0\rangle - b_j^*)\right) &\text{(B.12)}\\ &\geqslant \frac{\gamma}{2}, &\text{(B.13)}\end{aligned}$$

which implies

$$\sum_{j\in S(x)} \Delta_j = \frac{\sum_{j\in S(x)} \max_{t\in S(x)} a_t^*\Delta_j}{\max_{t\in S(x)} a_t^*} \geqslant \frac{\sum_{j\in S(x)} a_j^*\Delta_j}{\max_{t\in S(x)} a_t^*} \geqslant \frac{\gamma}{2\max_{t\in S(x)} a_t^*}. \qquad \text{(B.14)}$$

□

Lemma 3.5.12.

$$\left|\frac{Z^*_{\text{bulk}}(x,i)}{Z^*(x,i)} - \frac{Z_{\text{bulk}}(x,i)}{Z(x,i)}\right| \geqslant p_b(1-e^{-\varepsilon_p}). \tag{3.32}$$

Proof. If $\varepsilon_p = 0$, both sides of the inequality equal 0, and the inequality holds. When $\varepsilon_p > 0$,

$$\left|\frac{Z^*_{\text{bulk}}(x,i)}{Z^*(x,i)} - \frac{Z_{\text{bulk}}(x,i)}{Z(x,i)}\right| \tag{B.15}$$

$$= \frac{Z^*_{\text{bulk}}(x,i)}{Z^*(x,i)} \cdot \left|1 - \frac{Z_{\text{bulk}}(x,i)}{Z(x,i)} \cdot \frac{Z^*(x,i)}{Z^*_{\text{bulk}}(x,i)}\right| \tag{B.16}$$

$$= \frac{Z^*_{\text{bulk}}(x,i)}{Z^*(x,i)} \cdot \left|1 - \exp\left(\log\frac{Z_{\text{bulk}}(x,i)}{Z(x,i)} - \log\frac{Z^*_{\text{bulk}}(x,i)}{Z^*(x,i)}\right)\right| \tag{B.17}$$

$$\geqslant \frac{Z^*_{\text{bulk}}(x,i)}{Z^*(x,i)} \cdot \left|1 - \exp\left(-\left|\log\frac{Z_{\text{bulk}}(x,i)}{Z(x,i)} - \log\frac{Z^*_{\text{bulk}}(x,i)}{Z^*(x,i)}\right|\right)\right| \tag{B.18}$$

$$\geqslant p_b(1-e^{-\varepsilon_p}). \tag{B.19}$$

□

Lemma 3.5.13. *For all $j \in S(x)$,*

$$|p^*(x_i = j|x_{-i}) - p(x_i = j|x_{-i})| \geqslant e^{b^*_j - \varepsilon_b - \varepsilon_p} \cdot p_b \cdot \left(e^{\sigma(\Delta_j - 2\varepsilon_b - \varepsilon_p)} - 1\right). \tag{3.35}$$

Proof. We know that the word probabilities come from a log-linear model, so

$$|p^*(x_i = j|x_{-i}) - p(x_i = j|x_{-i})| = \left|\frac{\exp\left(\langle v^*_{-i}(x), v^*_j\rangle\right)}{Z^*(x,i)} - \frac{\exp(\langle v_{-i}(x), v_j\rangle)}{Z(x,i)}\right|.$$

If the probability $p(x_i = j|x_{-i})$ is reasonably large, i.e., its corresponding neuron is activated, then we can use that with Δ_j to bound this difference. In other words, when $\langle v_{-i}(x), v_j - v_0\rangle \geqslant b^*_j$, we know that

$$\left(\langle v_{-i}^*(x), v_j^*\rangle - \log Z^*\right) - \left(\langle v_{-i}(x), v_j\rangle - \log Z\right) \tag{B.20}$$

$$= \left(\langle v_{-i}^*(x), v_j^*\rangle - \log Z_{\text{bulk}}^*\right) - \left(\langle v_{-i}(x), v_j\rangle - \log Z_{\text{bulk}}\right) + \left(\log \frac{Z_{\text{bulk}}^*}{Z^*} - \log \frac{Z_{\text{bulk}}}{Z}\right) \tag{B.21}$$

$$\geqslant \left(\langle v_{-i}^*(x), v_j^*\rangle - \log Z_{\text{bulk}}^*\right) - \left(\langle v_{-i}(x), v_j\rangle - \log Z_{\text{bulk}}\right) - \varepsilon_p \tag{B.22}$$

$$= \langle v_{-i}^*(x), v_j^* - v_0^*\rangle - \langle v_{-i}(x), v_j - v_0\rangle \tag{B.23}$$

$$+ \left(\langle v_{-i}^*(x), v_0^*\rangle - \log Z_{\text{bulk}}^*\right) - \left(\langle v_{-i}(x), v_0\rangle - \log Z_{\text{bulk}}\right) - \varepsilon_p \tag{B.24}$$

$$\geqslant \Delta_j - 2\varepsilon_b - \varepsilon_p. \tag{B.25}$$

Note that the last inequality (B.25) comes from Assumption 3.5.5 and the following property of Δ_j:

$$\langle v_{-i}^*(x), v_j^* - v_0^*\rangle - \langle v_{-i}(x), v_j - v_0\rangle \tag{B.26}$$

$$= \langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^* - \left(\langle v_{-i}(x), v_j - v_0\rangle - b_j^*\right) \tag{B.27}$$

$$\geqslant \langle v_{-i}^*(x), v_j^* - v_0^*\rangle - b_j^* - \sigma\left(\langle v_{-i}(x), v_j - v_0\rangle - b_j^*\right) = \Delta_j. \tag{B.28}$$

Then we bound the difference in probabilities: for all $j \in S(x)$,

$$\left|\frac{\exp\left(\langle v_{-i}^*(x), v_j^*\rangle\right)}{Z^*} - \frac{\exp(\langle v_{-i}(x), v_j\rangle)}{Z}\right| \tag{B.29}$$

$$= \frac{\exp(\langle v_{-i}(x), v_j\rangle)}{Z} \cdot \left|\frac{\exp\left(\langle v_{-i}^*(x), v_j^*\rangle\right)}{Z^*} \cdot \frac{Z}{\exp(\langle v_{-i}(x), v_j\rangle)} - 1\right| \tag{B.30}$$

$$= \frac{\exp(\langle v_{-i}(x), v_j\rangle)}{Z} \cdot \left|\exp\left(\left(\langle v_{-i}^*(x), v_j^*\rangle - \log Z^*\right) - \left(\langle v_{-i}(x), v_j\rangle - \log Z\right)\right) - 1\right|. \tag{B.31}$$

Note that

$$\log \frac{\exp(\langle v_{-i}(x), v_j\rangle)}{Z} = \langle v_{-i}(x), v_j\rangle - \log Z \tag{B.32}$$

$$= \langle v_{-i}(x), v_j - v_0\rangle + \langle v_{-i}(x), v_0\rangle - \log Z_{\text{bulk}} + \log \frac{Z_{\text{bulk}}}{Z} \tag{B.33}$$

$$\geqslant b_j^* - \varepsilon_b + \log \frac{Z_{\text{bulk}}^*}{Z^*} + \left(\log \frac{Z_{\text{bulk}}}{Z} - \log \frac{Z_{\text{bulk}}^*}{Z^*}\right) \tag{B.34}$$

$$\geqslant b_j^* - \varepsilon_b + \log p_b - \varepsilon_p. \tag{B.35}$$

Moreover, when $\Delta_j - 2\varepsilon_b - \varepsilon_p > 0$, we have by (B.25) that

$$\left|\exp\left(\left(\langle v_{-i}^*(x), v_j^*\rangle - \log Z^*\right) - \left(\langle v_{-i}(x), v_j\rangle - \log Z\right)\right) - 1\right| \tag{B.36}$$

$$\geqslant \exp\left(\Delta_j - 2\varepsilon_b - \varepsilon_p\right) - 1. \tag{B.37}$$

When $\Delta_j - 2\varepsilon_b - \varepsilon_p \leqslant 0$, we have

$$\left|\exp\left(\left(\langle v_{-i}^*(x), v_j^*\rangle - \log Z^*\right) - \left(\langle v_{-i}(x), v_j\rangle - \log Z\right)\right) - 1\right| \geqslant 0 = e^0 - 1. \tag{B.38}$$

Thus,

$$\left|\exp\left(\left(\langle v_{-i}^*(x), v_j^*\rangle - \log Z^*\right) - \left(\langle v_{-i}(x), v_j\rangle - \log Z\right)\right) - 1\right| \geqslant e^{\sigma(\Delta_j - 2\varepsilon_b - \varepsilon_p)} - 1. \tag{B.39}$$

Therefore, when $\langle v_{-i}(x), v_j - v_0\rangle \geqslant b_j^*$, we must have

$$|p^*(x_i = j|x_{-i}) - p(x_i = j|x_{-i})| \geqslant e^{b_j^* - \varepsilon_b - \varepsilon_p} \cdot p_b \cdot \left(e^{\sigma(\Delta_j - 2\varepsilon_b - \varepsilon_p)} - 1\right). \tag{B.40}$$

In the second case where $\langle v_{-i}(x), v_j - v_0\rangle < b_j^*$, we have $\langle v_{-i}(x), v_j\rangle < \langle v_{-i}(x), v_0\rangle + b_j^*$. The proof for the second case is very similar to that of the first case, with the main difference being we replace $\langle v_{-i}(x), v_j\rangle$ in the first case by $\langle v_{-i}(x), v_0\rangle + b_j^*$ in the second case.

In the second case, by the definition of Δ_j we know that $\Delta_j = (\langle v^*_{-i}(x), v^*_j - v^*_0\rangle - b^*_j)$.

Similar to the first case, we can get

$$\big(\langle v^*_{-i}(x), v^*_j\rangle - \log Z^*\big) - \big(\langle v_{-i}(x), v_0\rangle + b^*_j - \log Z\big) \tag{B.41}$$

$$= \big(\langle v^*_{-i}(x), v^*_j\rangle - \log Z^*_{\text{bulk}}\big) - \big(\langle v_{-i}(x), v_0\rangle + b^*_j - \log Z_{\text{bulk}}\big) \tag{B.42}$$

$$+ \left(\log \frac{Z^*_{\text{bulk}}}{Z^*} - \log \frac{Z_{\text{bulk}}}{Z}\right) \tag{B.43}$$

$$\geqslant \big(\langle v^*_{-i}(x), v^*_j\rangle - \log Z^*_{\text{bulk}}\big) - \big(\langle v_{-i}(x), v_0\rangle + b^*_j - \log Z_{\text{bulk}}\big) - \varepsilon_p \tag{B.44}$$

$$\geqslant \langle v^*_{-i}(x), v^*_j - v^*_0\rangle - b^*_j + (\langle v^*_{-i}(x), v^*_0\rangle - \log Z^*_{\text{bulk}}) - (\langle v_{-i}(x), v_0\rangle - \log Z_{\text{bulk}}) - \varepsilon_p \tag{B.45}$$

$$\geqslant \Delta_j - 2\varepsilon_b - \varepsilon_p. \tag{B.46}$$

Therefore, for the difference in probabilities, if $\Delta_j - 2\varepsilon_b - \varepsilon_p > 0$, we have $\big(\langle v^*_{-i}(x), v^*_j\rangle - \log Z^*\big) > \big(\langle v_{-i}(x), v_0\rangle + b^*_j - \log Z\big)$, so

$$|p^*(x_i = j|x_{-i}) - p(x_i = j|x_{-i})| \tag{B.47}$$

$$= \left|\frac{\exp\big(\langle v^*_{-i}(x), v^*_j\rangle\big)}{Z^*} - \frac{\exp(\langle v_{-i}(x), v_j\rangle)}{Z}\right| \tag{B.48}$$

$$\geqslant \frac{\exp\big(\langle v^*_{-i}(x), v^*_j\rangle\big)}{Z^*} - \frac{\exp\big(\langle v_{-i}(x), v_0\rangle + b^*_j\big)}{Z} \tag{B.49}$$

$$= \frac{\exp\big(\langle v_{-i}(x), v_0\rangle + b^*_j\big)}{Z} \cdot \Big(\exp\big(\big(\langle v^*_{-i}(x), v^*_j\rangle - \log Z^*\big)\big) \tag{B.50}$$

$$- \big(\langle v_{-i}(x), v_0\rangle + b^*_j - \log Z\big)\big) - 1\Big) \tag{B.51}$$

$$\geqslant \frac{\exp\big(\langle v_{-i}(x), v_0\rangle + b^*_j\big)}{Z} \cdot \left(e^{\Delta_j - 2\varepsilon_b - \varepsilon_p} - 1\right). \tag{B.52}$$

If $\Delta_j - 2\varepsilon_b - \varepsilon_p \leqslant 0$, the absolute value of probability difference can be trivially

bounded below by 0. Thus,

$$|p^*(x_i = j|x_{-i}) - p(x_i = j|x_{-i})| \geqslant \frac{\exp\big(\langle v_{-i}(x), v_0\rangle + b_j^*\big)}{Z} \cdot \left(e^{\sigma(\Delta_j - 2\varepsilon_b - \varepsilon_p)} - 1\right). \tag{B.53}$$

Since

$$\log \frac{\exp\big(\langle v_{-i}(x), v_0\rangle + b_j^*\big)}{Z} = b_j^* + \langle v_{-i}(x), v_0\rangle - \log Z_{\text{bulk}} + \log \frac{Z_{\text{bulk}}}{Z} \tag{B.54}$$

$$\geqslant b_j^* - \varepsilon_b + \log \frac{Z_{\text{bulk}}^*}{Z^*} + \left(\log \frac{Z_{\text{bulk}}}{Z} - \log \frac{Z_{\text{bulk}}^*}{Z^*}\right) \tag{B.55}$$

$$\geqslant b_j^* - \varepsilon_b + \log p_b - \varepsilon_p, \tag{B.56}$$

we finally get the same bound as the first case:

$$|p^*(x_i = j|x_{-i}) - p(x_i = j|x_{-i})| \geqslant e^{b_j^* - \varepsilon_b - \varepsilon_p} \cdot p_b \cdot \left(e^{\sigma(\Delta_j - 2\varepsilon_b - \varepsilon_p)} - 1\right). \tag{B.57}$$

Merging the two cases, we know that for all $j \in S(x)$,

$$|p^*(x_i = j|x_{-i}) - p(x_i = j|x_{-i})| \geqslant e^{b_j^* - \varepsilon_b - \varepsilon_p} \cdot p_b \cdot \left(e^{\sigma(\Delta_j - 2\varepsilon_b - \varepsilon_p)} - 1\right). \tag{B.58}$$

This finishes the proof of Lemma 3.5.13. □

B.3 Experiment Details

Language models. We use various versions of GPT-2 Radford et al. (2019) and OPT Zhang et al. (2022b) with number of parameters ranging from 125M to 1.5B. We use the base, medium, large, xl version of GPT-2 which have #parameters 124M / 355M / 774M / 1.5B and hidden dimension 768 / 1024 / 1280 / 1600. For OPT, we use three versions, with #parameters 125M / 350M / 1.3B and hidden dimension 768 / 1024 / 2048. For all these models, the word probability is the softmax of the product of the penultimate layer representation and the dictionary. This is consistent

with our theoretical model introduced in Section 3.2. The parameter settings and performances of these models are shown in Table B.1.

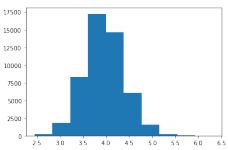
Table B.1: Parameters and Performances of Languge Models

Model	#Param	Hidden dim	Perp
GPT-2	124M	768	25.92
GPT-2 Medium	355M	1024	19.19
GPT-2 Large	774M	1280	17.13
GPT-2 XL	1.5B	1600	15.34
OPT-125M	125M	768	25.01
OPT-350M	350M	1024	19.69
OPT-1.3B	1.3B	2048	13.16

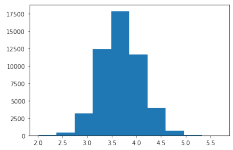
Dataset. We use WikiText-2 Merity et al. (2016) as the text corpus. WikiText-2 has about 280k tokens, and we only use the first 1/4 of it for computational efficiency. The perplexities of the language models on this corpus are shown in Table B.1.

Dictionary atoms are not uniformly distributed on sphere. Figure B.1 shows the histogram of the ℓ_2-norm of the dictionary atoms from different language models, and Figure B.2 shows the distribution of the Cosine similarity between two random atoms . The norms of these atoms are somewhat bounded, but their cosine similarity is strongly biased towards the positive part, and the dictionary matrices are close to low rank, indicating that these vectors are far from uniformly distributed on a sphere. Instead, they may concentrate around a cone-shaped region.

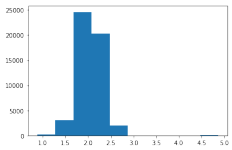
Log bulk partition function can be linearly approximated by hidden states. Table B.2 shows the mean squared approximation error of the log bulk partition function for different models and different k. For most of the models, the approximation error is small, and the approximation becomes better when k increases. GPT-2 is an



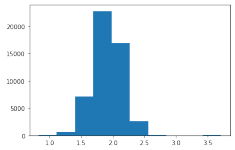
(a) GPT-2



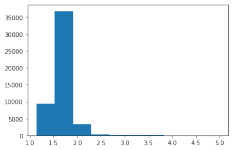
(b) GPT-2 Medium



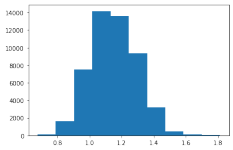
(c) GPT-2 Large



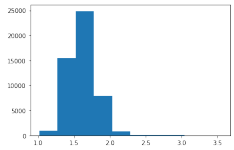
(d) GPT-2 XL



(e) OPT-125M

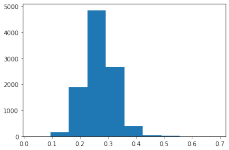


(f) OPT-350M

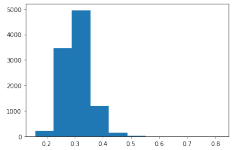


(g) OPT-1.3B

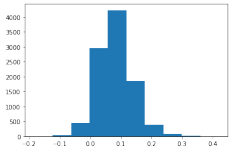
FIGURE B.1: ℓ_2-norms of atoms



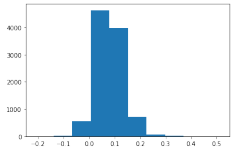
(a) GPT-2



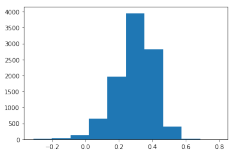
(b) GPT-2 Medium



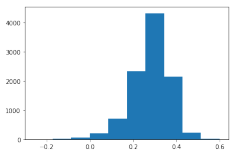
(c) GPT-2 Large



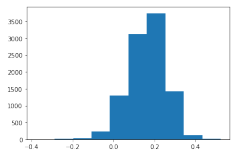
(d) GPT-2 XL



(e) OPT-125M

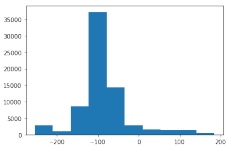


(f) OPT-350M

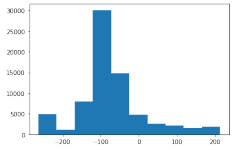


(g) OPT-1.3B

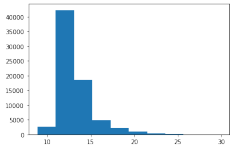
FIGURE B.2: Cosine Similarity between two random atoms



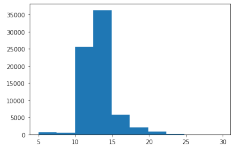
(a) GPT-2



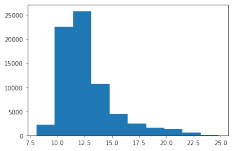
(b) GPT-2 Medium



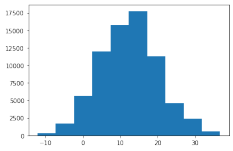
(c) GPT-2 Large



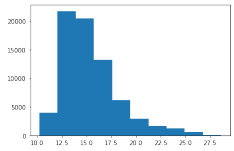
(d) GPT-2 XL



(e) OPT-125M

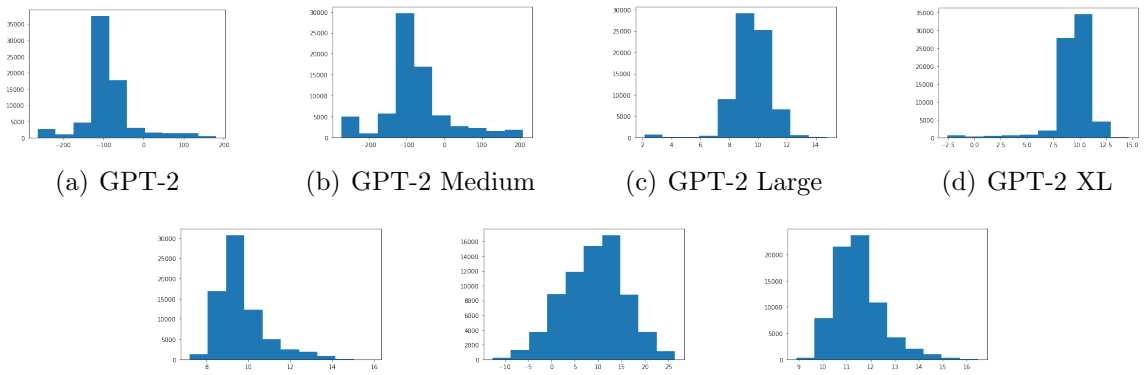


(f) OPT-350M



(g) OPT-1.3B

FIGURE B.3: Histogram of log partition function



(a) GPT-2 (b) GPT-2 Medium (c) GPT-2 Large (d) GPT-2 XL

(e) OPT-125M (f) OPT-350M (g) OPT-1.3B

FIGURE B.4: Histogram of log bulk partition function

exception to this rule: Its partition function form two clusters, each of which can be well approximated. In other words, there exits two anchor vectors for GPT-2.

Table B.2: Mean squared approximation error of log bulk partition function for different models and different k.

Model\k	0	10	100	1000	10000
GPT-2	625.8	651.4	665.2	685.5	727.4
GPT-2 M	0.7647	0.3444	0.3005	0.2683	0.2396
GPT-2 L	0.7296	0.1288	0.0740	0.0317	0.0022
GPT-2 XL	0.7716	0.1219	0.0648	0.0279	0.0019
OPT-125M	0.6688	0.1010	0.0443	0.0117	0.0006
OPT-350M	2.2449	0.9393	0.7587	0.6182	0.4275
OPT-1.3B	0.5234	0.0984	0.0402	0.0117	0.0006

Appendix C

Supplementary Materials for Chapter 4

C.1 Detailed Derivations

C.1.1 Derivation of Hessian

For an input $\boldsymbol{x}$ with label $\boldsymbol{y}$, we define the Hessian of single input loss with respect to vector $\boldsymbol{v}$ as

$$\boldsymbol{H}_\ell(\boldsymbol{v}, \boldsymbol{x}) = \nabla^2_{\boldsymbol{v}} \ell(f_\theta(\boldsymbol{x}), \boldsymbol{y}) = \nabla^2_{\boldsymbol{v}} \ell(\boldsymbol{z}_{\boldsymbol{x}}, \boldsymbol{y}). \tag{C.1}$$

We define the Hessian of loss with respect to $\boldsymbol{v}$ for the entire training sample as

$$\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{v}) = \nabla^2_{\boldsymbol{v}} \mathcal{L}(\theta) = \sum_{i=1}^{N} \nabla^2_{\boldsymbol{v}} \ell(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i) = \sum_{i=1}^{N} \boldsymbol{H}_\ell(\boldsymbol{v}, \boldsymbol{x}_i) = \mathbb{E}\left[\boldsymbol{H}_\ell(\boldsymbol{v}, \boldsymbol{x})\right]. \tag{C.2}$$

We now derive the Hessian for a fixed input label pair $(\boldsymbol{x}, \boldsymbol{y})$. Following the definition and notations in Section 4.2, we also denote output as $\boldsymbol{z} = f_\theta(\boldsymbol{x})$. We fix a layer p for the layer-wise Hessian. Here the layer-wise weight Hessian is $\boldsymbol{H}_\ell(\boldsymbol{w}^{(p)}, \boldsymbol{x})$. We also have the output for the layer as $\boldsymbol{z}^{(p)}$. Since $\boldsymbol{w}^{(p)}$ only appear in the layer but not the subsequent layers, we can consider $\boldsymbol{z} = f_\theta(\boldsymbol{x}) = g_\theta(\boldsymbol{z}^{(p)}(\boldsymbol{w}, \boldsymbol{x}))$ where g_θ only

contains the layers after the p-th layer and does not depend on $\boldsymbol{w}^{(p)}$. Thus, using the Hessian Chain rule (Skorski, 2019), we have

$$\boldsymbol{H}_\ell(\boldsymbol{w}^{(p)}, \boldsymbol{x}) = \left(\frac{\partial \boldsymbol{z}^{(p)}}{\partial \boldsymbol{w}^{(p)}}\right)^{\mathsf{T}} \boldsymbol{H}_\ell(\boldsymbol{z}^{(p)}, \boldsymbol{x}) \left(\frac{\partial \boldsymbol{z}^{(p)}}{\partial \boldsymbol{w}^{(p)}}\right) + \sum_{i=1}^{m^{(p)}} \frac{\partial \ell(\boldsymbol{z}, \boldsymbol{y})}{\partial z_i^{(p)}} \nabla^2_{\boldsymbol{w}^{(p)}} z_i^{(p)}, \tag{C.3}$$

where $z_i^{(p)}$ is the ith entry of $\boldsymbol{z}^{(p)}$ and $m^{(p)}$ is the number of neurons in p-th layer (size of $\boldsymbol{z}^{(p)}$).

Since $\boldsymbol{z}^{(p)} = \boldsymbol{W}^{(p)}\boldsymbol{x}^{(p)} + \boldsymbol{b}^{(p)}$ and $\boldsymbol{w}^{(p)} = \mathrm{vec}(\boldsymbol{W}^{(p)})$ we have

$$\frac{\partial \boldsymbol{z}^{(p)}}{\partial \boldsymbol{w}^{(p)}} = \boldsymbol{I}_{m^{(p)}} \otimes \boldsymbol{x}^{(p)\mathsf{T}}. \tag{C.4}$$

Since $\frac{\partial \boldsymbol{z}^{(p)}}{\partial \boldsymbol{w}^{(p)}}$ does not depend on $\boldsymbol{w}^{(p)}$, for all i we have $\nabla^2_{\boldsymbol{w}^{(p)}} z_i^{(p)} = 0$. Thus,

$$\boldsymbol{H}_\ell(\boldsymbol{w}^{(p)}, \boldsymbol{x}) = \left(\boldsymbol{I}_{m^{(p)}} \otimes \boldsymbol{x}^{(p)}\right) \boldsymbol{H}_\ell(\boldsymbol{z}^{(p)}, \boldsymbol{x}) \left(\boldsymbol{I}_{m^{(p)}} \otimes \boldsymbol{x}^{(p)\mathsf{T}}\right). \tag{C.5}$$

We define $\boldsymbol{M}_{\boldsymbol{x}}^{(p)} = \boldsymbol{H}_\ell(\boldsymbol{z}^{(p)}, \boldsymbol{x})$ as in Section 4.2 so that

$$\boldsymbol{H}_\ell(\boldsymbol{w}^{(p)}, \boldsymbol{x}) = \left(\boldsymbol{I}_{m^{(p)}} \otimes \boldsymbol{x}^{(p)}\right) \boldsymbol{M}_{\boldsymbol{x}}^{(p)} \left(\boldsymbol{I}_{m^{(p)}} \otimes \boldsymbol{x}^{(p)\mathsf{T}}\right) = \boldsymbol{M}_{\boldsymbol{x}}^{(p)} \otimes \boldsymbol{x}^{(p)}\boldsymbol{x}^{(p)\mathsf{T}}. \tag{C.6}$$

We now look into $\boldsymbol{M}_x^{(p)} = \boldsymbol{H}_\ell(\boldsymbol{z}^{(p)}, \boldsymbol{x})$. Again we have $\boldsymbol{z} = g_\theta(\boldsymbol{z}^{(p)})$ and can use chain rule here,

$$\boldsymbol{H}_\ell(\boldsymbol{z}^{(p)}, \boldsymbol{x}) = \left(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{z}^{(p)}}\right)^{\mathsf{T}} \boldsymbol{H}_\ell(\boldsymbol{z}, \boldsymbol{x}) \left(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{z}^{(p)}}\right) + \sum_{i=1}^{c} \frac{\partial \ell(\boldsymbol{z}, \boldsymbol{y})}{\partial z_i} \nabla^2_{\boldsymbol{z}^{(p)}} z_i \tag{C.7}$$

By letting $\boldsymbol{p} := \mathrm{softmax}(\boldsymbol{z})$ be the output confidence vector, we define the Hessian with respect to output logit $\boldsymbol{z}$ as $\boldsymbol{A}_{\boldsymbol{x}}$ and have

$$\boldsymbol{A}_{\boldsymbol{x}} := \boldsymbol{H}_\ell(\boldsymbol{z}, \boldsymbol{x}) = \nabla^2_{\boldsymbol{z}} l(\boldsymbol{z}, \boldsymbol{y}) = \mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^{\mathsf{T}}, \tag{C.8}$$

according to Singla et al. (2019).

We also define the Jacobian of $\boldsymbol{z}$ with respect to $\boldsymbol{z}^{(p)}$ (informally logit gradient for layer p) as $\boldsymbol{G}_{\boldsymbol{x}}^{(p)} := \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{z}^{(p)}}$. For FC layers with ReLUs, we can consider ReLU after the p-th layer as multiplying $\boldsymbol{z}^{(p)}$ by an indicator function $\mathbf{1}_{\boldsymbol{z}^{(p)}>0}$. To use matrix multiplication, we can turn the indicator function into a diagonal matrix and define it as $\boldsymbol{D}^{(p)}$ where

$$\boldsymbol{D}^{(p)} := \operatorname{diag}\left(\mathbf{1}_{\boldsymbol{z}^{(p)}>0}\right). \tag{C.9}$$

Thus, we have the input of the next layer as $\boldsymbol{x}^{(p+1)} = \boldsymbol{D}^{(p)}\boldsymbol{z}^{(p)}$. The FC layers can then be considered as a sequential matrix multiplication and we have the final output as

$$\boldsymbol{z} = \boldsymbol{W}^{(L)}\boldsymbol{D}^{(L-1)}\boldsymbol{W}^{(L-1)}\boldsymbol{D}^{(L-2)}\cdots\boldsymbol{D}^{(p)}\boldsymbol{z}^{(p)}. \tag{C.10}$$

Thus,

$$\boldsymbol{G}_{\boldsymbol{x}}^{(p)} = \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{z}^{(p)}} = \boldsymbol{W}^{(L)}\boldsymbol{D}^{(L-1)}\boldsymbol{W}^{(L-1)}\boldsymbol{D}^{(L-2)}\cdots\boldsymbol{D}^{(p)}. \tag{C.11}$$

Since $\boldsymbol{G}_{\boldsymbol{x}}^{(p)}$ is independent of $\boldsymbol{z}^{(p)}$, we have

$$\nabla^2_{\boldsymbol{z}^{(p)}} z_i = 0, \forall i. \tag{C.12}$$

Thus,

$$\boldsymbol{M}_{\boldsymbol{x}}^{(p)} = \boldsymbol{H}_\ell(\boldsymbol{z}^{(p)}, \boldsymbol{x}) = \boldsymbol{G}_{\boldsymbol{x}}^{(p)\mathsf{T}}\boldsymbol{A}_{\boldsymbol{x}}\boldsymbol{G}_{\boldsymbol{x}}^{(p)}. \tag{C.13}$$

Moreover, loss Hessian with respect to the bias term $\boldsymbol{b}^{(p)}$ equals to that with respect to the output of that layer $\boldsymbol{z}^{(p)}$. We thus have

$$\boldsymbol{H}_\ell(\boldsymbol{b}^{(p)}, \boldsymbol{x}) = \boldsymbol{M}_{\boldsymbol{x}}^{(p)} = \boldsymbol{G}_{\boldsymbol{x}}^{(p)\mathsf{T}}\boldsymbol{A}_{\boldsymbol{x}}\boldsymbol{G}_{\boldsymbol{x}}^{(p)}. \tag{C.14}$$

The Hessians of loss for the entire training sample are simply the empirical ex-

pectations of the Hessian for single input. We have the formula as the following:

$$\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)}) = \mathbb{E}\left[\boldsymbol{H}_{\ell}(\boldsymbol{w}^{(p)}, \boldsymbol{x})\right] = \mathbb{E}\left[\boldsymbol{M}_{\boldsymbol{x}}^{(p)} \otimes \boldsymbol{x}^{(p)}\boldsymbol{x}^{(p)\mathsf{T}}\right], \tag{C.15}$$

$$\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{b}^{(p)}) = \boldsymbol{H}_{\mathcal{L}}(\boldsymbol{z}^{(p)}) = \mathbb{E}\left[\boldsymbol{M}_{\boldsymbol{x}}^{(p)}\right] = \mathbb{E}\left[\boldsymbol{G}_{\boldsymbol{x}}^{(p)\mathsf{T}}\boldsymbol{A}_{\boldsymbol{x}}\boldsymbol{G}_{\boldsymbol{x}}^{(p)}\right]. \tag{C.16}$$

Note that we can further decompose $\boldsymbol{A}_{\boldsymbol{x}} = \boldsymbol{Q}_{\boldsymbol{x}}^{\mathsf{T}}\boldsymbol{Q}_{\boldsymbol{x}}$, where

$$\boldsymbol{Q}_{\boldsymbol{x}} = \operatorname{diag}\left(\sqrt{\boldsymbol{p}}\right)\left(\boldsymbol{I}_c - \mathbf{1}_c\boldsymbol{p}^{\mathsf{T}}\right), \tag{C.17}$$

with $\mathbf{1}_c$ is a all one vector of size c, proved in Papyan (2019).

We can further extend the close form expression to off diagonal blocks and the bias entries to get the full Gauss-Newton term of Hessian. Let

$$\boldsymbol{F}_{\boldsymbol{x}}^{\mathsf{T}} = \begin{pmatrix} \boldsymbol{G}_{\boldsymbol{x}}{}^{(1)\mathsf{T}} \otimes \boldsymbol{x}^{(1)} \\ \boldsymbol{G}_{\boldsymbol{x}}{}^{(1)\mathsf{T}} \\ \boldsymbol{G}_{\boldsymbol{x}}{}^{(2)\mathsf{T}} \otimes \boldsymbol{x}^{(2)} \\ \boldsymbol{G}_{\boldsymbol{x}}{}^{(2)\mathsf{T}} \\ \vdots \\ \boldsymbol{G}_{\boldsymbol{x}}{}^{(L)\mathsf{T}} \otimes \boldsymbol{x}^{(n)} \\ \boldsymbol{G}_{\boldsymbol{x}}{}^{(L)\mathsf{T}} \end{pmatrix}. \tag{C.18}$$

The full Hessian is given by

$$\boldsymbol{H}_{\mathcal{L}}(\theta) = \mathbb{E}\left[\boldsymbol{F}_{\boldsymbol{x}}^{\mathsf{T}}\boldsymbol{A}_{\boldsymbol{x}}\boldsymbol{F}_{\boldsymbol{x}}\right] + \mathbb{E}\left[\sum_{i=1}^{c}\frac{\partial\ell(\boldsymbol{z}, \boldsymbol{y})}{z_i}\nabla_{\theta}^2 z_i\right]. \tag{C.19}$$

C.1.2 Approximating Weight Hessian of Convolutional Layers

The approximation of weight Hessian of convolutional layer is a trivial extension from the approximation of Fisher information matrix of convolutional layer by Grosse and Martens (2016).

Consider a two dimensional convolutional layer of neural network with m input channels and n output channels. Let its input feature map $\mathbf{X}$ be of shape (n, X_1, X_2)

and output feature map $\mathbf{Z}$ be of shape (m, P_1, P_2). Let its convolution kernel be of size $K_1 \times K_2$. Then the weight $\mathbf{W}$ is of shape (m, n, K_1, K_2), and the bias $\boldsymbol{b}$ is of shape (m). Let P be the number of patches slide over by the convolution kernel, we have $P = P_1 P_2$.

Follow Dangel et al. (2020), we define $\boldsymbol{Z} \in \mathbb{R}^{m \times P}$ as the reshaped matrix of $\mathbf{Z}$ and $\boldsymbol{W} \in \mathbb{R}^{m \times nK_1K_2}$ as the reshaped matrix of $\mathbf{W}$. Define $\boldsymbol{B} \in \mathbb{R}^{m \times P}$ by broadcasting $\boldsymbol{b}$ to P dimensions. Let $\boldsymbol{X} \in \mathbb{R}^{nK_1K_2 \times P}$ be the unfolded $\mathbf{X}$ with respect to the convolutional layer. The unfold operation (Paszke et al., 2019b) is commonly used in computation to model convolution as matrix operations.

After the above transformation, we have the linear expression of the p-th convolutional layer similar to FC layers:

$$\boldsymbol{Z}^{(p)} = \boldsymbol{W}^{(p)} \boldsymbol{X}^{(p)} + \boldsymbol{B}^{(p)} \tag{C.20}$$

We still omit superscription of (p) for dimensions for simplicity. We also denote $\boldsymbol{z}^{(p)}$ as the vector form of $\boldsymbol{Z}^{(p)}$ and has size mP. Similar to fully connected layer, we have analogue of Equation C.6 for convolutional layer as

$$\boldsymbol{H}_\ell(\boldsymbol{w}^{(p)}, \boldsymbol{X}) = \left(\boldsymbol{I}_m \otimes \boldsymbol{X}^{(p)}\right) \boldsymbol{M}_{\boldsymbol{x}}^{(p)} \left(\boldsymbol{I}_m \otimes \boldsymbol{X}^{(p)\mathsf{T}}\right), \tag{C.21}$$

where $\boldsymbol{M}_{\boldsymbol{x}}^{(p)} = \boldsymbol{H}_\ell(\boldsymbol{z}^{(p)}, \boldsymbol{X})$ and is a $mP \times mP$ matrix. Also, since convolutional layers can also be considered as linear operations (matrix multiplication with reshape) together with FC layers and ReLUs, Equation C.12 still holds. Thus, we still have

$$\boldsymbol{H}_\ell(\boldsymbol{z}^{(p)}, \boldsymbol{X}) = \boldsymbol{M}_{\boldsymbol{x}}^{(p)} = \boldsymbol{G}_{\boldsymbol{x}}^{(p)\mathsf{T}} \boldsymbol{A}_{\boldsymbol{x}} \boldsymbol{G}_{\boldsymbol{x}}^{(p)}, \tag{C.22}$$

where $\boldsymbol{G}_{\boldsymbol{x}}^{(p)} = \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{z}^{(p)}}$ and has dimension $c \times mP$, although is cannot be further decomposed as direct multiplication of weight matrices as in the FC layers.

However, for convolutional layers, $\boldsymbol{X}^{(p)}$ is a matrix instead of a vector. Thus, we cannot make Equation C.21 into the form of a Kronecker product as in Equation C.6.

Despite this, it is still possible to have a Kronecker factorization of the weight Hessian in the form

$$\boldsymbol{H}_\ell(\boldsymbol{w}^{(p)}, \boldsymbol{X}) \approx \widetilde{\boldsymbol{M}}_{\boldsymbol{x}}^{(p)} \otimes \boldsymbol{X}^{(p)} \boldsymbol{X}^{(p)\mathsf{T}}, \tag{C.23}$$

using further approximation motivated by Grosse and Martens (2016). Note that $\widetilde{\boldsymbol{M}}_{\boldsymbol{x}}^{(p)}$ need to have a different shape ($m \times m$) from $\boldsymbol{M}_{\boldsymbol{x}}^{(p)}$ ($mP \times mP$), since $\boldsymbol{H}_\ell(\boldsymbol{w}^{(p)}, \boldsymbol{X})$ is $mnK1K2 \times mnK1K2$ and $\boldsymbol{X}^{(p)} \boldsymbol{X}^{(p)\mathsf{T}}$ is $nK1K2 \times nK1K2$.

Since we can further decompose $\boldsymbol{A}_{\boldsymbol{x}} = \boldsymbol{Q}_{\boldsymbol{x}}{}^{\mathsf{T}} \boldsymbol{Q}_{\boldsymbol{x}}$, we then have

$$\boldsymbol{M}_{\boldsymbol{x}}{}^{(p)} = \boldsymbol{G}_{\boldsymbol{x}}^{(p)\mathsf{T}} \boldsymbol{A}_{\boldsymbol{x}} \boldsymbol{G}_{\boldsymbol{x}}^{(p)} = \left(\boldsymbol{Q}_{\boldsymbol{x}} \boldsymbol{G}_{\boldsymbol{x}}{}^{(p)} \right)^{\mathsf{T}} \left(\boldsymbol{Q}_{\boldsymbol{x}} \boldsymbol{G}_{\boldsymbol{x}}{}^{(p)} \right). \tag{C.24}$$

We define $\boldsymbol{N}_{\boldsymbol{x}}^{(p)} = \boldsymbol{Q}_{\boldsymbol{x}} \boldsymbol{G}_{\boldsymbol{x}}{}^{(p)}$. Here $\boldsymbol{Q}_{\boldsymbol{x}}$ is $c \times c$ and $\boldsymbol{G}_{\boldsymbol{x}}{}^{(p)}$ is $c \times mP$ so that $\boldsymbol{N}_{\boldsymbol{x}}^{(p)}$ is $c \times mP$. We can reshape $\boldsymbol{N}_{\boldsymbol{x}}^{(p)}$ into a $cP \times m$ matrix $\widetilde{\boldsymbol{N}}_{\boldsymbol{x}}^{(p)}$. We then reduce $\boldsymbol{M}_{\boldsymbol{x}}^{(p)}$ ($mP \times mP$) into a $m \times m$ matrix as

$$\widetilde{\boldsymbol{M}}_{\boldsymbol{x}}^{(p)} = \frac{1}{P} \widetilde{\boldsymbol{N}}_{\boldsymbol{x}}^{(p)\mathsf{T}} \widetilde{\boldsymbol{N}}_{\boldsymbol{x}}^{(p)}. \tag{C.25}$$

The scalar $\frac{1}{P}$ is a normalization factor since we squeeze a dimension of size P into size 1.

Thus, we can have similar Kronecker factorization approximation as

$$\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)}) = \mathbb{E}\left[\boldsymbol{H}_\ell(\boldsymbol{w}^{(p)}, \boldsymbol{X}) \right] = \mathbb{E}\left[\left(\boldsymbol{I}_m \otimes \boldsymbol{X}^{(p)} \right) \boldsymbol{M}_{\boldsymbol{x}}^{(p)} \left(\boldsymbol{I}_m \otimes \boldsymbol{X}^{(p)\mathsf{T}} \right) \right] \tag{C.26}$$

$$\approx \mathbb{E}\left[\widetilde{\boldsymbol{M}}_{\boldsymbol{x}}^{(p)} \otimes \boldsymbol{X}^{(p)} \boldsymbol{X}^{(p)\mathsf{T}} \right] \approx \mathbb{E}\left[\widetilde{\boldsymbol{M}}_{\boldsymbol{x}}^{(p)} \right] \otimes \mathbb{E}\left[\boldsymbol{X}^{(p)} \boldsymbol{X}^{(p)\mathsf{T}} \right]. \tag{C.27}$$

C.2 Main Proof

This is the complete proof for the two main theorems sketched in Section 4.4.

C.2.1 Preliminaries

Notations

In this section, we generally follow the notation standard by Goodfellow et al. (2016). We will use bold italic lowercase letters ($\boldsymbol{v}$) to denote vectors, bold non-italic lowercase letters to denote random vectors ($\mathbf{v}$), bold italic uppercase letters ($\boldsymbol{A}$) to denote matrices, and bold italic uppercase letters ($\mathbf{A}$) to denote random matrices.

Moreover, we use $[n]$ for positive integer n to denote the set $\{1, \cdots, n\}$, and $\|\boldsymbol{M}\|$ to denote the spectral norm of a matrix $\boldsymbol{M}$. We use $\langle \boldsymbol{A}, \boldsymbol{B} \rangle_F$ to denote the Frobenius inner product of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, namely $\langle \boldsymbol{A}, \boldsymbol{B} \rangle_F \triangleq \sum_{i,j} \boldsymbol{A}_{i,j} \boldsymbol{B}_{i,j}$. We use $\mathrm{tr}(\boldsymbol{M})$ to denote the trace of a matrix $\boldsymbol{M}$, and we use $\mathbf{1}_c$ to denote the all-one vector of dimension c (the subscript may be omitted when it's clear from the context).

For probability distributions, we use $\mathcal{N}^{\mathrm{R}}(\mu, \sigma)$ to denote the rectified Gaussian distribution which has density function

$$f_{\mathcal{N}^{\mathrm{R}}}(x; \mu, \sigma) = \Phi\left(\frac{\mu}{\sigma}\right) \delta(x) + \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \mathbb{I}\left[x > 0\right]. \tag{C.28}$$

Here Φ is the CDF of standard normal distribution, $\delta(x)$ is the Dirac delta function. Note that when $\mu = 0$, the density function simplifies to

$$f_{\mathcal{N}^{\mathrm{R}}}(x; 0, \sigma) = \frac{1}{2}\delta(x) + \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \mathbb{I}\left[x > 0\right]. \tag{C.29}$$

We will use the same notation for multivariate rectified Gaussian distribution, which will be used to characterize the inputs of the network.

Problem Setting

Consider a two layer fully connected ReLU activated neural network with input dimension d, hidden layer dimension n and output dimension c. In particular, n goes to infinity, $d = n^{1+\alpha}$ for some $\alpha > 0$, and c is a finite constant. Let network be trained with cross-entropy objective $\mathcal{L}$. Let σ denote the element-wise ReLU activation function which acts as $\sigma(x) = x \cdot \mathbb{I}_{x \geqslant 0}$ and the product here is applied element-wise. Let $\boldsymbol{W}^{(1)} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{W}^{(2)} \in \mathbb{R}^{c \times n}$ denote the weight matrices of the first and second layer respectively.

We consider the case that the neural network has rectified standard Gaussian input $\mathbf{x} \sim \mathcal{N}^{\mathrm{R}}(0, \boldsymbol{I}_d)$. Denote the output of the first and second layer as $\mathbf{y}$ and $\mathbf{z}$ respectively. We have $\mathbf{y} = \sigma(\boldsymbol{W}^{(1)}\mathbf{x})$ and $\mathbf{z} = \boldsymbol{W}^{(2)}\mathbf{y}$. Let $\mathbf{p} = \mathrm{softmax}(\mathbf{z})$ denote the softmax output of the network and let $\mathbf{A} \triangleq \mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^{\mathsf{T}}$.

In this problem, we look into the state of random Gaussian initialization, in which entries of both matrices are i.i.d. sampled from a standard normal distribution, and then re-scaled such that each row of $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ has norm 1. When taking n and d to infinity, with the concentration of norm in high-dimensional Gaussian random variables, we assume in this problem that entries of $\boldsymbol{W}^{(1)}$ are iid sampled from a zero-mean distribution with variance $1/d$, and entries of $\boldsymbol{W}^{(2)}$ are iid sampled from a zero-mean distribution with variance $1/n$. This initialization is standard in training neural networks. From the previous analysis of Hessian, the output Hessian corresponding to the first layer has closed form

$$\boldsymbol{M}^{(1)} \triangleq \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{N}^{\mathrm{R}}(0, \boldsymbol{I}_d)} \left[\mathbf{D}\boldsymbol{W}^{(2)T}\mathbf{A}\boldsymbol{W}^{(2)}\mathbf{D} \right], \tag{C.30}$$

where $\mathbf{D} \triangleq \mathrm{diag}(\mathbb{I}\left[\mathbf{y} \geqslant 0\right]) \in \mathbb{R}^{n \times n}$ is the random 0/1 diagonal matrix representing

the activations of ReLU function after the first layer. Note that the output Hessian of the second layer is simply $\boldsymbol{M}^{(2)} \triangleq \mathbb{E}\left[\mathbf{A}\right]$.

By the Kronecker decomposition, the closed form of the layer-wise Hessians of the first and the second layer are

$$\boldsymbol{H}^{(1)} \triangleq \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{N}^{\mathrm{R}}(0,\boldsymbol{I}_d)}\left[\mathbf{D}\boldsymbol{W}^{(2)\mathsf{T}}\mathbf{A}\boldsymbol{W}^{(2)}\mathbf{D}\otimes\mathbf{x}\mathbf{x}^{\mathsf{T}}\right],$$

$$\boldsymbol{H}^{(2)} \triangleq \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{N}^{\mathrm{R}}(0,\boldsymbol{I}_d)}\left[\mathbf{A}\otimes\mathbf{D}\boldsymbol{W}^{(1)}\mathbf{x}\mathbf{x}^{\mathsf{T}}\boldsymbol{W}^{(1)\mathsf{T}}\mathbf{D}\right].$$

Following the decoupling conjecture, let the Kronecker approximation of the Hessians above be

$$\widehat{\boldsymbol{H}}^{(1)} \triangleq \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{N}^{\mathrm{R}}(0,\boldsymbol{I}_d)}\left[\mathbf{D}\boldsymbol{W}^{(2)\mathsf{T}}\mathbf{A}\boldsymbol{W}^{(2)}\mathbf{D}\right]\otimes\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{N}^{\mathrm{R}}(0,\boldsymbol{I}_d)}\left[\mathbf{x}\mathbf{x}^{\mathsf{T}}\right],$$

$$\widehat{\boldsymbol{H}}^{(2)} \triangleq \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{N}^{\mathrm{R}}(0,\boldsymbol{I}_d)}\left[\mathbf{A}\right]\otimes\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{N}^{\mathrm{R}}(0,\boldsymbol{I}_d)}\left[\mathbf{D}\boldsymbol{W}^{(1)}\mathbf{x}\mathbf{x}^{\mathsf{T}}\boldsymbol{W}^{(1)\mathsf{T}}\mathbf{D}\right].$$

The decoupling conjecture is then equivalent to $\boldsymbol{H}^{(1)} \approx \widehat{\boldsymbol{H}}^{(1)}$, $\boldsymbol{H}^{(2)} \approx \widehat{\boldsymbol{H}}^{(2)}$.

Since our formulae for the Hessians are going to depend on the weight matrices, throughout the section we will condition on the value of $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ when we take expectation (i.e. the expectation is only taken over the input $\mathbf{x} \sim \mathcal{N}^{\mathrm{R}}(0, \boldsymbol{I}_d)$). We will neglect this under-script of the expectation operator $\mathbb{E}$ as there will be no confusion. When we are discussing the Hessians of a certain layer, we will also neglect the upper-script and just use $\boldsymbol{H}$ and $\boldsymbol{M}$ when there is no confusion. Moreover, we denote $\mathbf{X} \triangleq \mathbb{E}\left[\mathbf{x}\mathbf{x}^{\mathsf{T}}\right]$ as the autocorrelation of the input.

Furthermore, for simplicity of notations, we will sometimes use the verbal description "with probability 1 over $\boldsymbol{W}^{(1)}/\boldsymbol{W}^{(2)}$, event E is true" to denote

$$\lim_{n\to\infty} \Pr_{\boldsymbol{W}^{(1)}\sim\mathcal{N}(0,\frac{1}{d}\boldsymbol{I}_{nd}),\boldsymbol{W}^{(2)}\sim\mathcal{N}(0,\frac{1}{n}\boldsymbol{I}_{cn})}\left[E\right] = 1. \tag{C.31}$$

C.2.2 Detailed Proof

First, we restate our main theorems:

Theorem 4.4.1 (Decoupling Theorem) *Let V_1 and V_2 be the top $c-1$ eigenspaces of $\boldsymbol{H}^{(1)}$ and $\widehat{\boldsymbol{H}}^{(1)}$ respectively, for all $\epsilon > 0$,*

$$\lim_{n\to\infty} \Pr_{\boldsymbol{W}^{(1)}\sim\mathcal{N}(0,\frac{1}{d}\boldsymbol{I}_{nd}),\boldsymbol{W}^{(2)}\sim\mathcal{N}(0,\frac{1}{n}\boldsymbol{I}_{cn})} \left[\text{Overlap}\left(V_1, V_2\right) > 1-\epsilon\right] = 1. \tag{C.32}$$

Moreover $\boldsymbol{H}^{(1)}$ has $c-1$ large eigenvalues that,

$$\lim_{n\to\infty} \Pr_{\boldsymbol{W}^{(1)}\sim\mathcal{N}(0,\frac{1}{d}\boldsymbol{I}_{nd}),\boldsymbol{W}^{(2)}\sim\mathcal{N}(0,\frac{1}{n}\boldsymbol{I}_{cn})} \left[\left(\frac{\lambda_c(\boldsymbol{H}^{(1)})}{\lambda_{c-1}(\boldsymbol{H}^{(1)})}\bigg|_{\boldsymbol{W}^{(1)},\boldsymbol{W}^{(2)}}\right) < \epsilon\right] = 1. \tag{C.33}$$

Theorem 4.4.2 *Let $\boldsymbol{M}^* \triangleq \mathbb{E}\left[\mathbf{D}'\boldsymbol{W}^{(2)\mathsf{T}}\mathbf{A}\boldsymbol{W}^{(2)}\mathbf{D}'\right]$ where $\mathbf{D}'$ is an independent copy of $\mathbf{D}$ and is independent of $\mathbf{A}$. Let S_1 and S_2 be the top $c-1$ eigenspaces of $\boldsymbol{M}^{(1)}$ and $\boldsymbol{M}^*$ respectively, for all $\epsilon > 0$,*

$$\lim_{n\to\infty} \Pr_{\boldsymbol{W}^{(1)}\sim\mathcal{N}(0,\frac{1}{d}\boldsymbol{I}_{nd}),\boldsymbol{W}^{(2)}\sim\mathcal{N}(0,\frac{1}{n}\boldsymbol{I}_{cn})} \left[\text{Overlap}\left(S_1, S_2\right) > 1-\epsilon\right] = 1. \tag{C.34}$$

Moreover,

$$\lim_{n\to\infty} \Pr_{\boldsymbol{W}^{(1)}\sim\mathcal{N}(0,\frac{1}{d}\boldsymbol{I}_{nd}),\boldsymbol{W}^{(2)}\sim\mathcal{N}(0,\frac{1}{n}\boldsymbol{I}_{cn})} \left[\left(\frac{\lambda_c(\boldsymbol{M})}{\lambda_{c-1}(\boldsymbol{M})}\bigg|_{\boldsymbol{W}^{(1)},\boldsymbol{W}^{(2)}}\right) < \epsilon\right] = 1. \tag{C.35}$$

Properties of Infinite Width Weight Matrices

We will first prove some simple properties of the Gaussian initialized weight matrices $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ that will facilitate our analysis. Recall that $\boldsymbol{W}^{(1)} \in \mathbb{R}^{d\times n}$ and $\boldsymbol{W}^{(2)} \in \mathbb{R}^{n\times c}$ where the output dimension c is a finite constant, the hidden layer width n goes

to infinity, and the input dimension $d = n^{1+\alpha}$ for some constant $\alpha > 0$.

Lemma C.2.1. *For all $i \in [c]$, for all $\epsilon > 0$,*

$$\lim_{n\to\infty} \Pr\left[\left|\sum_{j=1}^{n} \boldsymbol{W}_{ij}^{(2)}\right| \geqslant \epsilon\right] = 0. \tag{C.36}$$

Proof of Lemma C.2.1. Since each entry of $\boldsymbol{W}^{(2)}$ is initialized independently from $\mathcal{N}(0, \frac{1}{n})$, by Central Limit Theorem we have $\sum_{j=1}^{n} \boldsymbol{W}_{ij}^{(2)} \sim \mathcal{N}(0, \frac{1}{n})$. For any $\epsilon > 0$, fix ϵ. By Chebyshev's inequality,

$$\lim_{n\to\infty} \Pr\left[\left|\sum_{j=1}^{n} \boldsymbol{W}_{ij}^{(2)}\right| \geqslant \epsilon\right] < \lim_{n\to\infty} \frac{1}{n\epsilon^2} = 0. \tag{C.37}$$

□

Lemma C.2.2. *(Laurent and Massart, 2000) For $X \sim \chi_n^2$,*

$$\Pr\left[X - n \geqslant 2\sqrt{nt} + 2t\right] \leqslant e^{-t}, \qquad \Pr\left[X - n \leqslant -2\sqrt{nt}\right] \leqslant e^{-t}. \tag{C.38}$$

Lemma C.2.3. *For all $\epsilon > 0$,*

$$\lim_{n\to\infty} \Pr\left[\left|\left\|\boldsymbol{W}^{(2)}\right\|_F^2 - c\right| \geqslant \epsilon\right] = 0. \tag{C.39}$$

Beside, for all $i \in [c]$,

$$\lim_{n\to\infty} \Pr\left[\left|\left\|\boldsymbol{W}_i^{(2)}\right\|^2 - 1\right| \geqslant \epsilon\right] = 0. \tag{C.40}$$

Proof of Lemma C.2.3. For simplicity of notations, we will use $\boldsymbol{W}$ to denote $\boldsymbol{W}^{(2)}$ in this proof. Since each entry of $\boldsymbol{W}$ is initialized independently from $\mathcal{N}(0, \frac{1}{n})$, we know that $n\|\boldsymbol{W}\|_F^2 = \sum_{i=1}^{c}\sum_{j=1}^{n} n\boldsymbol{W}_{i,j}{}^2$ follows a χ_{cn}^2-distribution. From Lemma

C.2.2 we know that for large enough n,

$$\Pr\left[|n\|\boldsymbol{W}\|_F^2 - cn| \geqslant n\epsilon\right] \geqslant \Pr\left[|n\|\boldsymbol{W}\|_F^2 - cn| \geqslant 2\sqrt{c}n^{3/4} + 2n^{1/2}\right] \leqslant 2\exp\left(-n^{1/2}\right). \tag{C.41}$$

In other words,

$$\lim_{n\to\infty} \Pr\left[|\|\boldsymbol{W}\|_F^2 - c| \geqslant \epsilon\right] = \lim_{n\to\infty} \Pr\left[|n\|\boldsymbol{W}\|_F^2 - cn| \geqslant n\epsilon\right] = 0. \tag{C.42}$$

Similarly, for any $i \in [c]$, $n\|\boldsymbol{W}_i\|_F^2$ follows a χ_n^2-distribution, so for large enough n,

$$\Pr\left[|n\|\boldsymbol{W}_i\|_F^2 - n| \geqslant n\epsilon\right] \leqslant \Pr\left[|n\|\boldsymbol{W}\|_F^2 - n| \geqslant 2n^{3/4} + 2n^{1/2}\right] \leqslant 2\exp\left(-n^{1/2}\right), \tag{C.43}$$

which indicates that

$$\lim_{n\to\infty} \Pr\left[|\|\boldsymbol{W}_i\|^2 - 1| \geqslant \epsilon\right] = \lim_{n\to\infty} \Pr\left[|n\|\boldsymbol{W}_i\|^2 - n| \geqslant n\epsilon\right] = 0. \tag{C.44}$$

□

Lemma C.2.4. *Let $\boldsymbol{w}_i$ denote the i-th column vector of $\boldsymbol{W}^{(1)}$. With probability 1 over $\boldsymbol{W}^{(1)}$,*

$$\max_{i=1}^{d} \|\boldsymbol{w}_i\| < 5n^{-\frac{\alpha}{2}}. \tag{C.45}$$

Proof of Lemma C.2.4. Since entries of $\boldsymbol{W}^{(1)}$ are i.i.d. sampled from $\mathcal{N}(0, \frac{1}{n})$, each $\|\boldsymbol{w}_i\|^2$ obeys a χ_n^2 scaled by $\frac{1}{d} = n^{-(1+\alpha)}$. Thus by the tail bound of Lemma C.2.2, setting $t = n$ we have

$$\Pr\left[\|\boldsymbol{w}_i\|^2 \geqslant 5n^{-\alpha}\right] = \Pr\left[d\|\boldsymbol{w}_i\|^2 \geqslant n + 2\sqrt{n^2} + 2n\right] \leqslant e^{-n}. \tag{C.46}$$

By a Union bound we have

$$\Pr\left[\max_{i=1}^{d} \|\boldsymbol{w}_i\|^2 \geqslant 5n^{-\alpha}\right] \leqslant \sum_{i=1}^{d} \Pr\left[\|\boldsymbol{w}_i\|^2 \geqslant 5n^{-\alpha}\right] = de^{-n} = n^{1+\alpha}e^{-n}. \tag{C.47}$$

Since α is a constant, RHS converges to 0. Thus with probability 1 over $\boldsymbol{W}^{(1)}$, we have

$$\max_{i=1}^{d} \|\boldsymbol{w}_i\|^2 < 5n^{-\alpha}. \tag{C.48}$$

Taking square root on both sides completes the proof. □

Lemma C.2.5. *For any random matrix $\boldsymbol{W}$ For all $\epsilon > 0$,*

$$\lim_{n\to\infty} \Pr\left[\left\|\boldsymbol{W}^{(1)}\boldsymbol{W}^{(1)\mathsf{T}} - \boldsymbol{I}_c\right\| \geqslant \epsilon\right] = 0. \tag{C.49}$$

Besides, for all $i, j \in [c]$,

$$\lim_{n\to\infty} \Pr\left[|(\boldsymbol{W}^{(1)}\boldsymbol{W}^{(1)\mathsf{T}})_{i,j} - \delta_{i,j}| \geqslant \epsilon\right] = 0 \tag{C.50}$$

Here δ is the Kronecker delta function, i.e., $\delta_{i,j} = \mathbb{I}[i = j]$.

Proof of Lemma C.2.5. To prove this lemma we need the following tail bound:

Lemma C.2.6. *(Zhu, 2012) If S follows a Wishart distribution $\mathcal{W}_d(n, C)$, with $r = \mathrm{tr}(C)/\|C\|$, for $\theta \geqslant 0$ the following inequality holds that*

$$\Pr\left[\left\|\frac{1}{n}S - C\right\| \geqslant \left(\sqrt{\frac{2\theta(r+1)}{n}} + \frac{2\theta r}{n}\right)\|C\|\right] \leqslant 2d\exp(-\theta). \tag{C.51}$$

Since each entry of $\boldsymbol{W}^{(1)}$ is initialized independently from $\mathcal{N}(0, \frac{1}{d})$, we know that $\boldsymbol{W}^{(1)}\boldsymbol{W}^{(1)\mathsf{T}}$ follows Wishart distribution $\mathcal{W}_d(d, \frac{1}{d}\boldsymbol{I}_n)$. With $r = \mathrm{tr}(\frac{1}{d}\boldsymbol{I}_n)/\left\|\frac{1}{d}\boldsymbol{I}_n\right\| = n$

and set $\theta = n^{\frac{\alpha}{2}}$, from Lemma C.2.6, for $n \geqslant 1$ we get

$$\begin{aligned}
2d\exp\left(-n^{\frac{\alpha}{2}}\right) &\geqslant \Pr\left[\left\|\frac{1}{d}\boldsymbol{W}^{(1)}\boldsymbol{W}^{(1)\mathsf{T}} - \frac{1}{d}\boldsymbol{I}_n\right\| \geqslant \left(\sqrt{\frac{2\theta(n+1)}{d}} + \frac{2\theta n}{d}\right)\left\|\frac{1}{d}\boldsymbol{I}_n\right\|\right] \\
&= \Pr\left[\left\|\frac{1}{d}\boldsymbol{W}^{(1)}\boldsymbol{W}^{(1)\mathsf{T}} - \frac{1}{d}\boldsymbol{I}_n\right\| \geqslant \left(\sqrt{\frac{2n^{\frac{\alpha}{2}}(2n)}{n^{1+\alpha}}} + \frac{2n^{\frac{\alpha}{2}}n}{n^{1+\alpha}}\right)\left\|\frac{1}{d}\boldsymbol{I}_n\right\|\right] \\
&= \Pr\left[\left\|\boldsymbol{W}^{(1)}\boldsymbol{W}^{(1)\mathsf{T}} - \boldsymbol{I}_n\right\| \geqslant 2(n^{-\frac{\alpha}{4}} + n^{-\frac{\alpha}{2}})\right].
\end{aligned} \tag{C.52}$$

Fix any $\epsilon > 0$, we may find $N \in \mathbb{N}$ such that for all $n > N$, $2(n^{-\frac{\alpha}{4}} + n^{-\frac{\alpha}{2}}) < \epsilon$. For any $\epsilon' > 0$, we may find N' such that $2d\exp\left(-n^{\frac{\alpha}{2}}\right) = 2n^{1+\alpha}\exp\left(-n^{\frac{\alpha}{2}}\right) < \epsilon'$. Passing n to infinity we get

$$\lim_{n\to\infty} \Pr\left[\left\|\boldsymbol{W}^{(1)}\boldsymbol{W}^{(1)\mathsf{T}} - \boldsymbol{I}_n\right\| > \epsilon\right] = 0. \tag{C.53}$$

Then we proceed to analyze the entries. For all $i, j \in [n]$, we have

$$\begin{aligned}
\Pr\left[|(\boldsymbol{W}^{(1)}\boldsymbol{W}^{(1)\mathsf{T}})_{i,j} - \delta_{i,j}| \geqslant \epsilon\right] &\leqslant \Pr\left[\sum_{i,j=1}^{n} \Pr\left[(\boldsymbol{W}^{(1)}\boldsymbol{W}^{(1)\mathsf{T}})_{i,j} - \delta_{i,j}\right]^2 \geqslant \epsilon^2\right] \\
&= \Pr\left[\left\|\boldsymbol{W}^{(1)}\boldsymbol{W}^{(1)\mathsf{T}} - \boldsymbol{I}_n\right\|_F^2 \geqslant \epsilon^2\right] \\
&\leqslant \Pr\left[\left\|\boldsymbol{W}^{(1)}\boldsymbol{W}^{(1)\mathsf{T}} - \boldsymbol{I}_n\right\| \geqslant \frac{\epsilon}{\sqrt{n}}\right],
\end{aligned} \tag{C.54}$$

which implies that for all $i, j \in [n]$,

$$\lim_{n\to\infty} \Pr\left[|(\boldsymbol{W}^{(1)}\boldsymbol{W}^{(1)\mathsf{T}})_{i,j} - \delta_{i,j}| \geqslant \epsilon\right] = 0. \tag{C.55}$$

□

For the second weight matrix $\boldsymbol{W}^{(2)}$, where $\boldsymbol{W}^{(2)}\boldsymbol{W}^{(2)\mathsf{T}} \sim \mathcal{W}_c(n, \frac{1}{n}\boldsymbol{I}_c)$, we may

prove an identical statement as shown in the corollary below. The proof proceeds identical as above since we only need the ratio between the width and the height of $\boldsymbol{W}$, which is n/c in this case, to go to infinity.

Corollary C.2.7. *For all $\epsilon > 0$,*

$$\lim_{n\to\infty} \Pr\left[\left\|\boldsymbol{W}^{(2)}\boldsymbol{W}^{(2)\mathsf{T}} - \boldsymbol{I}_n\right\| \geqslant \epsilon\right] = 0. \tag{C.56}$$

Next we will establish the approximate equivalence between the scatter matrix $\boldsymbol{W}^{(2)\mathsf{T}}\boldsymbol{W}^{(2)}$ and the projection matrix $P_{\boldsymbol{W}^{(2)}}$.

Lemma C.2.8. *Let $P_{\boldsymbol{W}^{(2)}}$ be the projection matrix onto the row space of $\boldsymbol{W}^{(2)}$, then for all $\epsilon > 0$,*

$$\lim_{n\to\infty} \Pr\left[\left\|\boldsymbol{W}^{(2)\mathsf{T}}\boldsymbol{W}^{(2)} - P_{\boldsymbol{W}^{(2)}}\right\|_F^2 > \epsilon\right] = 0. \tag{C.57}$$

Proof of Lemma C.2.8. For simplicity of notations, in this proof we will neglect the layer index superscript and use $\boldsymbol{W}$ to denote $\boldsymbol{W}^{(2)}$. Recall that $\boldsymbol{W} \in \mathbb{R}^{n\times c}$.

Fix $\epsilon \in (0,1)$ without loss of generality. Let $\boldsymbol{W}_i (i \in [c])$ be the i-th row of $\boldsymbol{W}$, and we will do the Gram–Schmidt process for the rows of $\boldsymbol{W}$. Specifically, the Gram–Schmidt process is as following: Assume that the basis $\{\overline{\boldsymbol{W}}_i\}_{i=1}^k$ are already normalized, we set $\boldsymbol{W}'_{k+1} \triangleq \boldsymbol{W}_{k+1} - \sum_{i=1}^k \langle \boldsymbol{W}_{k+1}, \overline{\boldsymbol{W}}_i\rangle$ and $\overline{\boldsymbol{W}}_{k+1} \triangleq \boldsymbol{W}'_{k+1}/\left\|\boldsymbol{W}'_{k+1}\right\|$. Finally, from the definition of projection matrix, we know that $P_{\boldsymbol{W}} = \overline{\boldsymbol{W}}^{\mathsf{T}}\overline{\boldsymbol{W}}$.

From Lemma C.2.3 we have for all $i \in [c]$,

$$\lim_{n\to\infty} \Pr\left[\left|\|\boldsymbol{W}_i\|^2 - 1\right| \geqslant \epsilon\right] = 0. \tag{C.58}$$

Let $\epsilon' \triangleq \epsilon^2/\left(c^3 \cdot 16^{2c+1}\right)$, from Lemma C.2.5 we know that for all $i, j \in [c]$,

$$\lim_{n\to\infty} \Pr\left[\left|\boldsymbol{W}_i\boldsymbol{W}_j^{\mathsf{T}} - \delta_{i,j}\right| \geqslant \epsilon'\right] = 0. \tag{C.59}$$

Then we use induction to bound the difference between $\boldsymbol{W}$ and $\overline{\boldsymbol{W}}$. Specifically, we will show that for all $i \in [c], \left\|\overline{\boldsymbol{W}}_i - \boldsymbol{W}_i\right\| \leqslant 8^i\epsilon'$. For simplicity of notations, in the following proof we will not repeat the probability argument and assume that for all $i, j \in [c]$, $|\boldsymbol{W}_i\boldsymbol{W}_j^{\mathsf{T}} - \delta_{i,j}| \leqslant \epsilon'$ and for all $i \in [c]$, $|\|\boldsymbol{W}_i\|^2 - 1| \leqslant \epsilon'$. We will only use these inequalities finite times so applying a union bound will give the probability result.

For $i = 1$, we know that $\overline{\boldsymbol{W}}_1 = \boldsymbol{W}_1/\|\boldsymbol{W}_1\|$ and $|\|\boldsymbol{W}_1\| - 1| \leqslant \epsilon'$, so $\left\|\overline{\boldsymbol{W}}_i - \boldsymbol{W}_i\right\| \leqslant \epsilon'$.

If our inductive hypothesis holds for $i \leqslant k$, then for $i = k + 1$, we have for all $j \leqslant k$,

$$\begin{aligned}
|\langle \boldsymbol{W}_i, \overline{\boldsymbol{W}}_j\rangle| &\leqslant |\langle \boldsymbol{W}_i, \boldsymbol{W}_j\rangle| + |\langle \boldsymbol{W}_i, \overline{\boldsymbol{W}}_j - \boldsymbol{W}_j\rangle| \\
&\leqslant \epsilon' + \|\boldsymbol{W}_i\| \cdot \left\|\overline{\boldsymbol{W}}_j - \boldsymbol{W}_j\right\| \\
&\leqslant \epsilon' + (1 + \epsilon')8^j\epsilon' \\
&\leqslant (2^{3j+1} + 1)\epsilon'.
\end{aligned} \tag{C.60}$$

Therefore,

$$\|\boldsymbol{W}_i' - \boldsymbol{W}_i\| \leqslant \sum_{j\in[k]} |\langle \boldsymbol{W}_i, \overline{\boldsymbol{W}}_j\rangle| \leqslant \epsilon' + \sum_{j\in[k]} (2^{3j+1} + 1)\epsilon' \leqslant (2^{3k+2} - 1)\epsilon', \tag{C.61}$$

and

$$|\|\boldsymbol{W}_i'\| - 1| \leqslant |\|\boldsymbol{W}_i\| - 1| + \|\boldsymbol{W}_i' - \boldsymbol{W}_i\| \leqslant 2^{3k+2}\epsilon'. \tag{C.62}$$

Thus,

$$\begin{aligned}
\left\|\overline{\boldsymbol{W}}_i - \boldsymbol{W}_i\right\| &\leqslant \left\|\overline{\boldsymbol{W}}_i - \boldsymbol{W}_i'\right\| + \|\boldsymbol{W}_i' - \boldsymbol{W}_i\| \\
&\leqslant |\|\boldsymbol{W}_i'\| - 1| + \|\boldsymbol{W}_i' - \boldsymbol{W}_i\| \\
&\leqslant 8^{k+1}\epsilon',
\end{aligned} \tag{C.63}$$

which finishes the induction and implies that for all $\epsilon > 0$, for all $i \in [c]$, $\left\|\overline{\boldsymbol{W}}_i - \boldsymbol{W}_i\right\| \leqslant 8^i\epsilon'$. Thus,

$$\left\|\overline{\boldsymbol{W}} - \boldsymbol{W}\right\|_F^2 = \sum_{i\in[c]} \left\|\overline{\boldsymbol{W}}_i - \boldsymbol{W}_i\right\|^2 \leqslant c \cdot 16^c\epsilon'. \tag{C.64}$$

This means that

$$\begin{aligned}\left\|\boldsymbol{W}^\mathsf{T}\boldsymbol{W} - P_{\boldsymbol{W}}\right\|_F &= \left\|\boldsymbol{W}^\mathsf{T}\boldsymbol{W} - \overline{\boldsymbol{W}}^\mathsf{T}\overline{\boldsymbol{W}}\right\|_F \\ &\leqslant 2\left\|\boldsymbol{W} - \overline{\boldsymbol{W}}\right\|_F\left\|\overline{\boldsymbol{W}}\right\|_F + \left\|\boldsymbol{W} - \overline{\boldsymbol{W}}\right\|_F^2 \\ &\leqslant 2c \cdot \sqrt{c} \cdot 8^c\sqrt{\epsilon'} + c \cdot 16^c\epsilon' \leqslant \epsilon.\end{aligned} \tag{C.65}$$

□

For the final property of the weight matrices, we show that the maximum among all entry of the weight matrices are reasonably small with high probability.

Lemma C.2.9. *Fix any $\alpha > 0$, consider $\boldsymbol{W} \in \mathbb{R}^{a\times b}$ for some $b > a^{1+\alpha}$ such that each entry is sampled from a zero mean Gaussian $\mathcal{N}(0, \frac{1}{b})$. The largest entry of $\boldsymbol{W}$ is reasonably small with high probability as b goes to infinity, namely,*

$$\lim_{b\to\infty} \Pr\left[\max_{(i,j)\in[a]\times[b]} |\boldsymbol{W}_{ij}^{(2)}| > 2b^{-\frac{1}{3}}\right] = 0 \tag{C.66}$$

Proof of Lemma C.2.9. For i.i.d. random variables $\mathbf{x}_1, \ldots, \mathbf{x}_b \sim \mathcal{N}(0, 1)$, by concentration inequality on maximum of Gaussian random variables, for any $t > 0$, we have

$$\Pr\left[\max_{i\in[b]} \mathbf{x}_i > \sqrt{2\log(2b)} + t\right] < 2e^{-\frac{t^2}{2}}. \tag{C.67}$$

For any $i, j \in [a] \times [b]$, since $\boldsymbol{W}_{ij}$ are i.i.d. sampled from $\mathcal{N}(0, \frac{1}{b})$, with rescaling of

$1/\sqrt{b}$ we may substitute $\mathbf{x}_j$ with $\boldsymbol{W}_{ij}$. It follows that

$$\Pr\left[\max_{(i,j)\in[a]\times[b]} \boldsymbol{W}_{ij}^{(2)} > \frac{\sqrt{2\log(2ab)}+t}{\sqrt{b}}\right] < 2e^{-\frac{t^2}{2}}. \tag{C.68}$$

Taking $t = b^{\frac{1}{6}}$, since $a < b$, for large b we have $\sqrt{2\log(2ab)} < \sqrt{2\log(2b^2)} < b^{\frac{1}{6}}$. Thus for large b,

$$\begin{aligned}\Pr\left[\max_{(i,j)\in[a]\times[b]} \boldsymbol{W}_{ij} > 2b^{-\frac{1}{3}}\right] &= \Pr\left[\max_{(i,j)\in[a]\times[b]} \boldsymbol{W}_{ij} > \frac{b^{\frac{1}{6}}+b^{\frac{1}{6}}}{\sqrt{n}}\right] \\ &< \Pr\left[\max_{(i,j)\in[a]\times[b]} \boldsymbol{W}_{ij} > \frac{\sqrt{2\log(2b)}+b^{\frac{1}{6}}}{\sqrt{b}}\right] < 2e^{-\frac{b^{\frac{1}{3}}}{2}}.\end{aligned} \tag{C.69}$$

With the same argument, we have

$$\Pr\left[\min_{(i,j)\in[a]\times[b]} \boldsymbol{W}_{ij} < -2b^{-\frac{1}{3}}\right] < 2e^{-\frac{b^{\frac{1}{3}}}{2}}. \tag{C.70}$$

Passing b to infinity completes the proof. □

From the above lemma, we can bound the maximum entry of $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ as follows:

Corollary C.2.10. *With probability 1 over $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$,*

$$\begin{aligned}\lim_{n\to\infty} \Pr\left[\max_{(i,j)\in[n]\times[d]} |\boldsymbol{W}_{ij}^{(1)}| > 2d^{-\frac{1}{3}}\right] = 0, \\ \lim_{n\to\infty} \Pr\left[\max_{(i,j)\in[c]\times[n]} |\boldsymbol{W}_{ij}^{(2)}| > 2n^{-\frac{1}{3}}\right] = 0.\end{aligned} \tag{C.71}$$

Approximate Independence Between Layer Inputs and Outputs

Let us first recall some definitions and notations of the inputs and outputs of layers. The input $\mathbf{x}$ follows the d-dimensional multivariate rectified Gaussian distribution

with identity covariance for the pre-rectified Gaussian, namely $\mathbf{x} \sim \mathcal{N}^{\mathrm{R}}(0, \boldsymbol{I}_d)$. The input propagates through the first layer to $\mathbf{u} \triangleq \boldsymbol{W}^{(1)}\mathbf{x}$, and is multiplied element-wise by the ReLU activation to the input of the second layer $\mathbf{y} \triangleq \sigma(\mathbf{u})$. Here we denote that activation of ReLU function by the random matrix $\mathbf{D} \triangleq \mathrm{diag}(\mathbb{I}[\mathbf{u} \geqslant 0]) \in \mathbb{R}^{n\times n}$. Finally we get the logit output of the network $\mathbf{z} \triangleq \boldsymbol{W}^{(2)}\mathbf{y}$. The output Hessian of the last layer is $\mathbf{A} = \mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^{\mathsf{T}} \in \mathbb{R}^{c\times c}$.

In this section we will show that when n goes to infinity, both $\mathbf{y}$ and $\mathbf{z}$ will converge in distribution to rectified Gaussian. Moreover, when we condition on two entries of $\mathbf{x}$ and two entries of $\mathbf{y}$, the output Hessian $\mathbf{A}$ will be invariant in the limiting case.

Lemma C.2.11. *When $d \to \infty$, with probability 1 over $\boldsymbol{W}^{(1)}$,*

$$\lim_{d\to\infty} \mathbf{y} \xrightarrow{d} \mathcal{N}^{\mathrm{R}}\left(0, \frac{\pi-1}{2\pi}\boldsymbol{I}_n\right).$$

Proof of Lemma C.2.11. We will prove this lemma using the multivariate Lindeberg-Feller CLT. Given that x_i's are i.i.d. sampled from $\mathcal{N}^{\mathrm{R}}(0,1)$ with bounded moments:

$$\mathbb{E}[\mathrm{x}_i] = \frac{1}{\sqrt{2\pi}},\ \mathbb{E}[(\mathrm{x}_i - \mathbb{E}[\mathrm{x}_i])^2] = \frac{\pi-1}{2\pi},\ \mathbb{E}[(\mathrm{x}_i - \mathbb{E}[\mathrm{x}_i])^4] = \frac{6\pi^2-10\pi-3}{4\pi^2} < 1. \tag{C.72}$$

For each $i \in [d]$, let $\boldsymbol{w}_i^{(1)} \in \mathbb{R}^d$ denote the i-th column vector of $\boldsymbol{W}^{(1)}$. Let $\mathbf{s}_i = \boldsymbol{w}_i^{(1)}(\mathrm{x}_i - \mathbb{E}[\mathrm{x}_i])$, then we have

$$\mathbf{y} = \sum_{i=1}^{d} \boldsymbol{w}_i^{(1)}\mathrm{x}_i = \sum_{i=1}^{d} \mathbf{s}_i + \sum_{i=1}^{d} \mathbb{E}[\mathrm{x}_i]\boldsymbol{w}_i^{(1)} = \sum_{i=1}^{d} \mathbf{s}_i + \frac{1}{\sqrt{2\pi}}\sum_{i=1}^{d} \boldsymbol{w}_i^{(1)}. \tag{C.73}$$

It follows that

$$Var[\mathbf{s}_i] = Var[\boldsymbol{w}_i^{(1)}\mathrm{x}_i] = \frac{\pi-1}{2\pi}\boldsymbol{w}_i^{(1)}\boldsymbol{w}_i^{(1)\mathsf{T}}. \tag{C.74}$$

Let $\boldsymbol{S} = \sum_{i=1}^{d} Var[\mathbf{s}_i]$,

$$\boldsymbol{S} = \frac{\pi - 1}{2\pi} \sum_{i=1}^{d} \boldsymbol{w}_i^{(1)} \boldsymbol{w}_i^{(1)\mathsf{T}} = \frac{\pi - 1}{2\pi} \boldsymbol{W}^{(1)} \boldsymbol{W}^{(1)\mathsf{T}}. \tag{C.75}$$

As $d \to \infty$, from Corollary C.2.7 we have $\boldsymbol{W}^{(1)} \boldsymbol{W}^{(1)\mathsf{T}} \to \boldsymbol{I}_n$ in probability, therefore

$$\lim_{d \to \infty} \boldsymbol{S} = \frac{\pi - 1}{2\pi} \boldsymbol{I}_n. \tag{C.76}$$

We now verify the Lindeberg condition of independent random vectors $\{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$. First observe that the fourth moments of the $\mathbf{s}_i$'s are sufficiently small.

$$\begin{aligned}
\lim_{d \to \infty} \sum_{i=1}^{d} \mathbb{E}\left[\|\mathbf{s}_i\|^4\right] &= \lim_{d \to \infty} \sum_{i=1}^{d} \mathbb{E}\left[\left(\sum_{j=1}^{n} \left(\boldsymbol{W}_{ji}^{(1)} \left(\mathrm{x}_i - \mathbb{E}[\mathrm{x}_i]\right)\right)^2\right)^2\right] \\
&\leqslant \lim_{d \to \infty} \sum_{i=1}^{d} \mathbb{E}\left[c^2 \left(\left(\max_{j \in [n]} \boldsymbol{W}_{ji}^{(1)}\right)^2 \left(\mathrm{x}_i - \mathbb{E}[\mathrm{x}_i]\right)^2\right)^2\right] \\
&\leqslant \lim_{d \to \infty} c^2 \left(\max_{i \in [d], j \in [n]} \boldsymbol{W}_{ji}^{(1)}\right)^4 \sum_{i=1}^{d} \mathbb{E}\left[\left(\mathrm{x}_i - \mathbb{E}[\mathrm{x}_i]\right)^4\right].
\end{aligned} \tag{C.77}$$

Since $\mathbb{E}[(\mathrm{x}_i - \mathbb{E}[\mathrm{x}_i])^4] < 1$ and $\max_{i \in [d], j \in [n]} |\boldsymbol{W}_{ji}^{(1)}| < 2d^{-\frac{1}{3}}$, with probability 1 over $\boldsymbol{W}^{(1)}$ from Lemma C.2.9, it follows that

$$\lim_{d \to \infty} \sum_{i=1}^{d} \mathbb{E}\left[\|\mathbf{s}_i\|^4\right] \leqslant c^2 \lim_{d \to \infty} \left(2d^{-\frac{1}{3}}\right)^4 \sum_{i=1}^{d} 1 = c^2 \lim_{d \to \infty} 16 d^{-\frac{4}{3}} n = 16 c^2 \lim_{d \to \infty} d^{-\frac{1}{3}} = 0. \tag{C.78}$$

For any $\epsilon > 0$, since $\|\mathbf{s}_i\| > \epsilon$ in the domain of integration (when $\mathbb{I}[\|\mathbf{s}_i\| > \epsilon]$),

$$\begin{aligned}\lim_{d\to\infty}\sum_{i=1}^{d}\mathbb{E}\left[\|\mathbf{s}_i\|^2\,\mathbb{I}\left[\|\mathbf{s}_i\| > \epsilon\right]\right] &< \lim_{d\to\infty}\sum_{i=1}^{d}\mathbb{E}\left[\frac{\|\mathbf{s}_i\|^2}{\epsilon^2}\,\|\mathbf{s}_i\|^2\,\mathbb{I}\left[\|\mathbf{s}_i\| > \epsilon\right]\right] \\ &\leqslant \frac{1}{\epsilon^2}\lim_{d\to\infty}\sum_{i=1}^{d}\mathbb{E}\left[\|\mathbf{s}_i\|^4\right] = 0.\end{aligned} \tag{C.79}$$

As the Lindeberg Condition is satisfied, with $\lim_{d\to\infty}\boldsymbol{S} = \frac{\pi-1}{2\pi}\boldsymbol{I}_n$ we have

$$\lim_{d\to\infty}\sum_{i=1}^{d}\mathbf{s}_i \xrightarrow{d} \mathcal{N}\left(0, \frac{\pi-1}{2\pi}\boldsymbol{I}_n\right). \tag{C.80}$$

By Lemma C.2.1, we have $\lim_{d\to\infty}\boldsymbol{w}_i^{(1)} = \overrightarrow{0}$ with probability 1 over $\boldsymbol{W}^{(1)}$, therefore plugging Equation C.80 into Equation C.73 we have

$$\lim_{d\to\infty}\mathbf{y} \xrightarrow{d} \mathcal{N}\left(0, \frac{\pi-1}{2\pi}\boldsymbol{I}_n\right). \tag{C.81}$$

Which completes the proof. □

Lemma C.2.12. $\lim_{n\to\infty}\mathbf{z} \xrightarrow{d} \mathcal{N}(0, \frac{(\pi-1)^2}{4\pi^2}\boldsymbol{I}_c)$ *with probability 1 over* $\boldsymbol{W}^{(2)}$.

Proof of Lemma C.2.12. The proof technique for $\mathbf{z}$ is identical to that of $\mathbf{y}$. For completeness we will redo it for $\boldsymbol{W}^{(2)}$. From Lemma C.2.11, y_i's are i.i.d. from $\mathcal{N}^{\mathrm{R}}(0, \frac{\pi-1}{2\pi})$ with bounded moments:

$$\mathbb{E}[\mathrm{y}_i] = \frac{\sqrt{\pi-1}}{2\pi},\ \mathbb{E}[(\mathrm{y}_i - \mathbb{E}[\mathrm{y}_i])^2] = \frac{(\pi-1)^2}{4\pi^2}, \tag{C.82}$$

$$\mathbb{E}[(\mathrm{y}_i - \mathbb{E}[\mathrm{y}_i])^4] = \frac{(6\pi^2-10\pi-3)(\pi-1)}{8\pi^3} < 1. \tag{C.83}$$

For each $i \in [n]$, let $\boldsymbol{w}_i^{(2)} \in \mathbb{R}^c$ denote the i-th column vector of $\boldsymbol{W}^{(2)}$. Let $\mathbf{v}_i =$

$\boldsymbol{w}_i^{(2)}(\mathrm{y}_i - \mathbb{E}[\mathrm{y}_i])$, then we have

$$\mathbf{z} = \sum_{i=1}^{n} \boldsymbol{w}_i^{(2)} \mathrm{y}_i = \sum_{i=1}^{n} \mathbf{v}_i + \sum_{i=1}^{n} \mathbb{E}[\mathrm{y}_i] \boldsymbol{w}_i^{(2)} = \sum_{i=1}^{n} \mathbf{v}_i + \frac{\sqrt{\pi - 1}}{2\pi} \sum_{i=1}^{n} \boldsymbol{w}_i^{(2)}. \tag{C.84}$$

It follows that

$$Var[\mathbf{v}_i] = Var[\boldsymbol{w}_i^{(2)} \mathrm{y}_i] = \frac{(\pi - 1)^2}{4\pi^2} \boldsymbol{w}_i^{(2)} \boldsymbol{w}_i^{(2)\mathsf{T}}. \tag{C.85}$$

Let $\boldsymbol{V} = \sum_{i=1}^{n} Var[\mathbf{v}_i]$,

$$\boldsymbol{V} = \frac{(\pi - 1)^2}{4\pi^2} \sum_{i=1}^{n} \boldsymbol{w}_i^{(2)} \boldsymbol{w}_i^{(2)\mathsf{T}} = \frac{(\pi - 1)^2}{4\pi^2} \boldsymbol{W}^{(2)} \boldsymbol{W}^{(2)\mathsf{T}}. \tag{C.86}$$

As $n \to \infty$, from Corollary C.2.7 we have $\boldsymbol{W}^{(2)} \boldsymbol{W}^{(2)\mathsf{T}} \to \boldsymbol{I}_c$ in probability, therefore

$$\lim_{n \to \infty} \boldsymbol{V} = \frac{(\pi - 1)^2}{4\pi^2} \boldsymbol{I}_c. \tag{C.87}$$

We now verify the Lindeberg condition of independent random vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$. First observe that the fourth moments of the $\mathbf{v}_i$'s are sufficiently small.

$$\begin{aligned}
\lim_{n \to \infty} \sum_{i=1}^{n} \mathbb{E}\left[\|\mathbf{v}_i\|^4\right] &= \lim_{n \to \infty} \sum_{i=1}^{n} \mathbb{E}\left[\left(\sum_{j=1}^{c} \left(\boldsymbol{W}_{ji}^{(2)} \left(\mathrm{y}_i - \mathbb{E}[\mathrm{y}_i]\right)\right)^2\right)^2\right] \\
&\leqslant \lim_{n \to \infty} \sum_{i=1}^{n} \mathbb{E}\left[c^2 \left(\left(\max_{j \in [c]} \boldsymbol{W}_{ji}^{(2)}\right)^2 \left(\mathrm{y}_i - \mathbb{E}[\mathrm{y}_i]\right)^2\right)^2\right] \\
&\leqslant \lim_{n \to \infty} c^2 \left(\max_{i \in [n], j \in [c]} \boldsymbol{W}_{ji}^{(2)}\right)^4 \sum_{i=1}^{n} \mathbb{E}\left[\left(\mathrm{y}_i - \mathbb{E}[\mathrm{y}_i]\right)^4\right].
\end{aligned} \tag{C.88}$$

Since $\mathbb{E}[(\mathrm{y}_i - \mathbb{E}[\mathrm{y}_i])^4] < 1$ and $\max_{i \in [n], j \in [c]} |\boldsymbol{W}_{ji}^{(2)}| < 2n^{-\frac{1}{3}}$ with probability 1 from

Corollary C.2.10, it follows that

$$\lim_{n\to\infty}\sum_{i=1}^{n}\mathbb{E}\left[\|\mathbf{v}_i\|^4\right] \leqslant c^2\lim_{n\to\infty}\left(2n^{-\frac{1}{3}}\right)^4\sum_{i=1}^{n}1 = c^2\lim_{n\to\infty}16n^{-\frac{4}{3}}n = 16c^2\lim_{n\to\infty}n^{-\frac{1}{3}} = 0. \tag{C.89}$$

For any $\epsilon > 0$, since $\|\mathbf{v}_i\| > \epsilon$ in the domain of integration (when $\mathbb{I}[\|\mathbf{v}_i\| > \epsilon]$),

$$\begin{aligned}\lim_{n\to\infty}\sum_{i=1}^{n}\mathbb{E}\left[\|\mathbf{v}_i\|^2\,\mathbb{I}\left[\|\mathbf{v}_i\| > \epsilon\right]\right] &< \lim_{n\to\infty}\sum_{i=1}^{n}\mathbb{E}\left[\frac{\|\mathbf{v}_i\|^2}{\epsilon^2}\|\mathbf{v}_i\|^2\,\mathbb{I}\left[\|\mathbf{v}_i\| > \epsilon\right]\right] \\ &\leqslant \frac{1}{\epsilon^2}\lim_{n\to\infty}\sum_{i=1}^{n}\mathbb{E}\left[\|\mathbf{v}_i\|^4\right] = 0.\end{aligned} \tag{C.90}$$

As the Lindeberg Condition is satisfied, with $\lim_{n\to\infty}\boldsymbol{V} = \frac{(\pi-1)^2}{4\pi^2}\boldsymbol{I}_c$ we have

$$\lim_{n\to\infty}\sum_{i=1}^{n}\mathbf{v}_i \xrightarrow{d} \mathcal{N}\left(0, \frac{(\pi-1)^2}{4\pi^2}\boldsymbol{I}_c\right). \tag{C.91}$$

By Lemma C.2.1, we have $\lim_{n\to\infty}\boldsymbol{w}_i^{(2)} = \overrightarrow{0}$ with probability 1 over $\boldsymbol{W}^{(2)}$, therefore plugging Equation C.91 into Equation C.84 we have

$$\lim_{n\to\infty}\mathbf{z} \xrightarrow{d} \mathcal{N}\left(0, \frac{(\pi-1)^2}{4\pi^2}\boldsymbol{I}_c\right). \tag{C.92}$$

Which completes the proof. □

Now we will show a key lemma for proving the main theorem, which suggests that when reasonably conditioning on two entries of the input $\mathbf{x}$ and two entries of the activation $\mathbf{D}$, the distribution of $\mathbf{z}$ converges in distribution to $\mathbf{z}$ without conditioning as $n \to \infty$.

Lemma C.2.13. *With probability 1 over $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$, fix any $\beta < \frac{\alpha}{2}$ (recall that $d = n^{1+\alpha}$), fix any $a, b \in (-n^\beta, n^\beta)$, for any $p, q \in [n]$ and $k, l \in [d]$, we have the*

following convergence in distribution

$$\mathbf{z}|(\mathbf{D}_{pp}=1,\mathbf{D}_{qq}=1,\mathbf{x}_k=a,\mathbf{x}_l=b)\xrightarrow{d}\mathbf{z}. \tag{C.93}$$

Proof of Lemma C.2.13. For simplicity of notation, we will use subscript $|_{\mathbf{x}}$ and $|_{\mathbf{D}}$ to denote the conditions we impose. For example, we will denote $\mathbf{z}|(\mathbf{D}_{pp}=1,\mathbf{D}_{qq}=1,\mathbf{x}_k=a,\mathbf{x}_l=b)$ by $\mathbf{z}|_{\mathbf{D},\mathbf{x}}$, and denote $\mathbf{x}|(\mathbf{x}_k=a,\mathbf{x}_l=b)$ by $\mathbf{x}|_{\mathbf{x}}$ etc.

First claim that with probability 1 over $\boldsymbol{W}^{(1)}$, $\mathbf{u}$ is invariant upon the conditioning on $\mathbf{x}$. Let $\boldsymbol{e}^{(i)}\in\mathbb{R}^d$ be the standard basis vector such that $\boldsymbol{e}^{(i)}_j=\mathbb{I}[i=j]$. Then

$$\begin{aligned}
\|\mathbf{u}-\mathbf{u}|_{\mathbf{x}}\| &= \left\|\boldsymbol{W}^{(1)}\mathbf{x}-\boldsymbol{W}^{(1)}\mathbf{x}|_{\mathbf{x}}\right\| \\
&= \left\|\boldsymbol{W}^{(1)}(\mathbf{x}-\mathbf{x}|_{\mathbf{x}})\right\| \\
&= \left\|\boldsymbol{W}^{(1)}((\mathbf{x}_k-a)\boldsymbol{e}^{(k)}+(\mathbf{x}_l-b)\boldsymbol{e}^{(l)})\right\| \\
&= \left\|\boldsymbol{w}^{(1)}_k\right\||\mathbf{x}_k-a|+\left\|\boldsymbol{w}^{(1)}_l\right\||\mathbf{x}_l-b| \\
&\leqslant 5n^{-\frac{\alpha}{2}}(|\mathbf{x}_k|+|\mathbf{x}_l|+|a|+|b|) \\
&\leqslant 5n^{-\frac{\alpha}{2}}(\mathbf{x}_k+\mathbf{x}_l)+10n^{-\frac{\alpha}{2}}n^{\frac{\beta}{2}}.
\end{aligned} \tag{C.94}$$

The norms of $\boldsymbol{w}^{(1)}_k$ and $\boldsymbol{w}^{(1)}_l$ are bounded from Lemma C.2.4. Note that as $n\to\infty$ we have $n^{-\frac{\alpha}{2}}$ and $n^{-\frac{\alpha-\beta}{2}}$ converging to 0 as we set $\beta<\alpha$. Since $\mathbf{x}$ is of bounded expectation and variance, $5n^{-\frac{\alpha}{2}}(\mathbf{x}_k+\mathbf{x}_l)$ converges in distribution to 0. Therefore $\|\mathbf{u}-\mathbf{u}|_{\mathbf{x}}\|\xrightarrow{d}0$ and hence $\mathbf{y}|_{\mathbf{x}}\xrightarrow{d}\mathbf{y}$. Since $\mathbf{z}$ is determined by $\mathbf{y}$, to prove $\mathbf{z}|_{\mathbf{D},\mathbf{x}}\xrightarrow{d}\mathbf{z}$, we now only ne ed to show $\mathbf{z}|_{\mathbf{D}}\xrightarrow{d}\mathbf{z}$.

Note that conditioning on $\mathbf{D}_{pp}=\mathbf{D}_{qq}=1$ is equivalent to conditioning on $\mathbf{u}_p>0$ and $\mathbf{u}_q>0$. Which is again equivalent to conditioning on $\mathbf{y}_p$ and $\mathbf{y}_q$ to be a half Gaussian distribution truncated at 0 instead of the rectified Gaussian. Recall that $\mathbf{z}=\boldsymbol{W}^{(2)}\mathbf{y}=\sum_{i=1}^n\boldsymbol{w}^{(2)}_i\mathbf{y}_i$. Since only $\mathbf{y}_p$ and $\mathbf{y}_q$ are affected by conditioning on $\mathbf{D}$,

we have

$$\begin{aligned}\|\mathbf{z}-\mathbf{z}|_{\mathbf{D}}\| &= \left\|\sum_{i=1}^{n}\boldsymbol{w}_i^{(2)}\mathbf{y}_i-\sum_{i=1}^{n}\boldsymbol{w}_i^{(2)}(\mathbf{y}|_{\mathbf{D}})_i\right\| \\ &= \left\|\boldsymbol{w}_p^{(2)}(\mathbf{y}_p-(\mathbf{y}|_{\mathbf{D}})_p)+\boldsymbol{w}_q^{(2)}(\mathbf{y}_q-(\mathbf{y}|_{\mathbf{D}})_q)\right\| \\ &\leqslant \left\|\boldsymbol{w}_p^{(2)}\right\||\mathbf{y}_p-(\mathbf{y}|_{\mathbf{D}})_p|+\left\|\boldsymbol{w}_q^{(2)}\right\||\mathbf{y}_q-(\mathbf{y}|_{\mathbf{D}})_q|.\end{aligned} \tag{C.95}$$

Note that $\mathbf{y}_p-(\mathbf{y}|_{\mathbf{D}})_p$ and $\mathbf{y}_q-(\mathbf{y}|_{\mathbf{D}})_q$ are difference between a rectified Gaussian with finite variance and its corresponding truncated Gaussian, both are of bounded expectation and variance. Meanwhile, by Lemma C.2.10, for all $i\in[n]$ we have that with probability 1 over $\boldsymbol{W}^{(2)}$,

$$\left\|\boldsymbol{w}_i^{(2)}\right\|\leqslant\sqrt{c\left(\max_{i\in[c],j\in[n]}\boldsymbol{W}_{ij}^{(2)}\right)^2}<\sqrt{4cn^{-\frac{2}{3}}}. \tag{C.96}$$

Since $\lim_{n\to\infty}\sqrt{4cn^{-\frac{2}{3}}}=0$, as n goes to infinity we have

$$\left\|\boldsymbol{w}_p^{(2)}\right\||\mathbf{y}_p-(\mathbf{y}|_{\mathbf{D}})_p|+\left\|\boldsymbol{w}_q^{(2)}\right\||\mathbf{y}_q-(\mathbf{y}|_{\mathbf{D}})_q|\xrightarrow{d}\overrightarrow{0}. \tag{C.97}$$

Therefore $\mathbf{z}|_{\mathbf{D}}\xrightarrow{d}\mathbf{z}$, and hence

$$\mathbf{z}|(\mathbf{D}_{pp}=1,\mathbf{D}_{qq}=1,\mathbf{x}_k=a,\mathbf{x}_l=b)\xrightarrow{d}\mathbf{z}. \tag{C.98}$$

□

Given that $\mathbf{p}=\mathrm{softmax}(\mathbf{z})$ and $\mathbf{A}=\mathrm{diag}(\mathbf{p})-\mathbf{p}\mathbf{p}^{\mathsf{T}}$, the mapping from $\mathbf{z}$ to $\mathbf{A}$ is bounded and continuous. Thus by the Portmanteau Theorem, we have the following corollary,

Corollary C.2.14. *For any $\epsilon>0$, with probability 1 over $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$, fix any $\beta<\frac{\alpha}{2}$ (recall that $d=n^{1+\alpha}$), fix any $a,b\in(-n^{\beta},n^{\beta})$, for any $p,q\in[n]$, $k,l\in[d]$,*

and $i, j \in [c]$*, we have*

$$|\mathbb{E}\left[\mathbf{A}_{ij} | (\mathbf{D}_{pp} = 1, \mathbf{D}_{qq} = 1, \mathbf{x}_k = a, \mathbf{x}_l = b)\right] - \mathbb{E}\left[\mathbf{A}_{ij}\right]| < \epsilon. \tag{C.99}$$

By the proof of Lemma C.2.13, this property holds when dropping the conditioning on $\mathbf{D}$ *or* $\mathbf{x}$*.*

Structure of $\mathbf{A}$

In this section we will analyze properties of the second output Hessian $\mathbf{A}$, which, despite being a $\mathbb{R}^{c\times c}$ "small" matrix, provides many important properties to the first output Hessian and the full layer-wise Hessians.

Lemma C.2.15. *With probability 1 over* $\boldsymbol{W}^{(1)}$ *and* $^{(2)}$*,* $\widetilde{\mathbf{A}} \triangleq \lim_{n\to\infty} \mathbb{E}[\mathbf{A}]$ *exist and is rank-*$(c-1)$ *.*

Proof of Lemma C.2.15. Note that each entry of $\mathbf{A}$ is a quadratic function of $\mathbf{p}$, and $\mathbf{p}$ is a continuous function of $\mathbf{z}$. Therefore, we consider $\mathbf{A}$ as a function of $\mathbf{z}$ and write $\mathbf{A}(\mathbf{z})$ when necessary. From Lemma C.2.12 we know that $\lim_{n\to\infty} \mathbf{z}$ follows a standard normal distribution $\mathcal{N}(0, \gamma \boldsymbol{I}_c)$ with probability 1 over $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$, where γ is some absolute constant. Therefore, $\widetilde{\mathbf{A}} \triangleq \lim_{n\to\infty} \mathbb{E}[\mathbf{A}]$ exist and it equals $\mathbb{E}[\mathbf{A}(\lim_{n\to\infty} \mathbf{z})] = \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(0,\gamma\boldsymbol{I}_c)}[\mathbf{A}(\mathbf{z})]$. For simplicity of notations, we will omit the statement "with probability 1 over $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$" when there is no confusion.

From the definition of $\mathbf{A}$ we know that $\mathbf{A} \triangleq \operatorname{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\mathsf{T}$ where $\mathbf{p}$ is the vector obtained by applying softmax to $\mathbf{z}$, so $\sum_{i=1}^c \mathbf{p}_i = 1$ and for all $i \in [c], \mathbf{p}_i \in (0, 1)$. Therefore, for any vector $\mathbf{p}$ satisfying the previous conditions, we have

$$\mathbf{1}^\mathsf{T}\mathbf{A}\mathbf{1} = \sum_{i=1}^c \left(\mathbf{p}_i - \sum_{j=1}^c \mathbf{p}_i\mathbf{p}_j\right) = \sum_{i=1}^c (\mathbf{p}_i - \mathbf{p}_i) = 0, \tag{C.100}$$

where $\mathbf{1}$ is the all-one vector. Therefore, we know that $\mathbf{A}$ has an eigenvalue 0 with eigenvector $c^{-\frac{1}{2}}\mathbf{1}$. This means that $\mathbb{E}[\mathbf{A}]$ also has an eigenvalue 0 with eigenvector $c^{-\frac{1}{2}}\mathbf{1}$. Thus, $\mathbb{E}[\mathbf{A}]$ is at most of rank $(c-1)$.

Then we analyze the other $(c-1)$ eigenvalues of $\widetilde{\mathbf{A}}$. Since $\mathbf{A} = \mathbf{Q}\mathbf{Q}^\mathsf{T}$ where $\mathbf{Q} = \operatorname{diag}(\sqrt{\mathbf{p}})(\boldsymbol{I}_c - \mathbf{1}\mathbf{p}^\mathsf{T})$, we know that $\mathbf{A}$ is always a positive semi-definite (PSD) matrix, which indicates that $E[\mathbf{A}]$ must also be PSD. Assume the c eigenvalues of $\widetilde{\mathbf{A}}$ are $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_{c-1} \geqslant \lambda_c = 0$. Therefore, by definition, we have

$$\lambda_{c-1} = \min_{\boldsymbol{v}\in S, \|\boldsymbol{v}\|=1} \boldsymbol{v}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{v} = \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{N}(0,\gamma\boldsymbol{I}_c)}\left[\min_{\boldsymbol{v}\in S, \|\boldsymbol{v}\|=1} \boldsymbol{v}^\mathsf{T}\mathbf{A}\boldsymbol{v}\right], \tag{C.101}$$

where $S \triangleq \mathbb{R}^c \backslash \mathcal{R}\{\mathbf{1}^\mathsf{T}\}$ is the orthogonal subspace of the span of $\mathbf{1}$. $\boldsymbol{v} \in S$ implies that $\boldsymbol{v} \perp \mathbf{1}$, i.e., $\sum_{i=1}^c \boldsymbol{v}_i = 0$.

Direct computation gives us

$$\boldsymbol{v}^\mathsf{T}\mathbf{A}\boldsymbol{v} = \sum_{i=1}^c \boldsymbol{v}_i^2 \mathbf{p}_i - \left(\sum_{i=1}^c \boldsymbol{v}_i \mathbf{p}_i\right)^2. \tag{C.102}$$

Define two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^c$ as for all $i \in [c]$, with $\boldsymbol{a}_i \triangleq \boldsymbol{v}_i\sqrt{\mathbf{p}_i}, \boldsymbol{b}_i \triangleq \sqrt{\mathbf{p}_i}$, then $\|\boldsymbol{b}\|^2 = \sum_{i=1}^c \mathbf{p}_i = 1$ and

$$\boldsymbol{v}^\mathsf{T}\mathbf{A}\boldsymbol{v} = \|\boldsymbol{a}\|^2 - \langle \boldsymbol{a}, \boldsymbol{b}\rangle^2 = \|\boldsymbol{a}\|^2 \cdot \|\boldsymbol{b}\|^2 - \langle \boldsymbol{a}, \boldsymbol{b}\rangle^2. \tag{C.103}$$

Therefore,

$$\boldsymbol{v}^\mathsf{T}\mathbf{A}\boldsymbol{v} \geqslant \|\boldsymbol{a}\|^2\|\boldsymbol{b}\|^2 \sin^2\theta(\boldsymbol{a}, \boldsymbol{b}), \tag{C.104}$$

where $\theta(\boldsymbol{a}, \boldsymbol{b})$ is the angle between $\boldsymbol{a}$ and $\boldsymbol{b}$, i.e., $\theta(\boldsymbol{a}, \boldsymbol{b}) \triangleq \arccos\frac{\langle \boldsymbol{a}, \boldsymbol{b}\rangle}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|}$. Define

$\mathbf{p}_0 \triangleq \min_{i\in[c]} \mathbf{p}_i$, then

$$\|\boldsymbol{a}\|^2 = \sum_{i=1}^{c} \boldsymbol{v}_i^2 \mathbf{p}_i \geqslant \sum_{i=1}^{c} \boldsymbol{v}_i^2 \mathbf{p}_0 = \mathbf{p}_0 \|\boldsymbol{v}\|^2 = \mathbf{p}_0. \tag{C.105}$$

Since $\|\boldsymbol{b}\| = 1$, we have

$$\sin^2\theta(\boldsymbol{a},\boldsymbol{b}) = \frac{\|\boldsymbol{a} - \langle \boldsymbol{a},\boldsymbol{b}\rangle \cdot \boldsymbol{b}\|^2}{\|\boldsymbol{a}\|^2}. \tag{C.106}$$

Besides,

$$\begin{aligned}
\|\boldsymbol{a} - \langle \boldsymbol{a},\boldsymbol{b}\rangle \cdot \boldsymbol{b}\|^2 &= \sum_{i=1}^{c} \left(\boldsymbol{v}_i \sqrt{\mathbf{p}_i} - \left(\sum_{j=1}^{c} \boldsymbol{v}_j \mathbf{p}_j \right) \sqrt{\mathbf{p}_i} \right)^2 \\
&= \sum_{i=1}^{c} \mathbf{p}_i \left(\boldsymbol{v}_i - \sum_{j=1}^{c} \boldsymbol{v}_j \mathbf{p}_j \right)^2 \\
&\geqslant \mathbf{p}_0 \sum_{i=1}^{c} \left(\boldsymbol{v}_i - \sum_{j=1}^{c} \boldsymbol{v}_j \mathbf{p}_j \right)^2.
\end{aligned} \tag{C.107}$$

Define $s \triangleq \arg\max_{i\in[c]} \boldsymbol{v}_i$ and $t \triangleq \arg\min_{i\in[c]} \boldsymbol{v}_i$, then

$$\sum_{i=1}^{c} \left(\boldsymbol{v}_i - \sum_{j=1}^{c} \boldsymbol{v}_j \mathbf{p}_j \right)^2 \geqslant \left(\boldsymbol{v}_s - \sum_{j=1}^{c} \boldsymbol{v}_j \mathbf{p}_j \right)^2 + \left(\boldsymbol{v}_t - \sum_{j=1}^{c} \boldsymbol{v}_j \mathbf{p}_j \right)^2 \geqslant \frac{(\boldsymbol{v}_s - \boldsymbol{v}_t)^2}{2}. \tag{C.108}$$

From $\|\boldsymbol{v}\| = 1$ we know that $\max_{i\in[c]} |\boldsymbol{v}_i| \geqslant c^{-\frac{1}{2}}$. Besides, since $\sum_{i=1}^{c} \boldsymbol{v}_i = 0$, we have $\boldsymbol{v}_s > 0 > \boldsymbol{v}_t$. Therefore, $\boldsymbol{v}_s - \boldsymbol{v}_t > \max_{i\in[c]} |\boldsymbol{v}_i| \geqslant c^{-\frac{1}{2}}$. As a result,

$$\|\boldsymbol{a} - \langle \boldsymbol{a},\boldsymbol{b}\rangle \cdot \boldsymbol{b}\|^2 \geqslant \mathbf{p}_0 \cdot \frac{(\boldsymbol{v}_s - \boldsymbol{v}_t)^2}{2} > \frac{\mathbf{p}_0}{2c}. \tag{C.109}$$

Moreover,

$$\|\boldsymbol{a}\|^2 = \sum_{i=1}^{c} \boldsymbol{v}_i^2 \mathbf{p}_i \leqslant \sum_{i=1}^{c} \mathbf{p}_i = 1. \tag{C.110}$$

Thus,

$$\sin^2 \theta(\boldsymbol{a}, \boldsymbol{b}) \geqslant \frac{\frac{\mathbf{p}_0}{2c}}{1} = \frac{\mathbf{p}_0}{2c}, \tag{C.111}$$

which means that

$$\boldsymbol{v}^\top \mathbf{A} \boldsymbol{v} \geqslant \mathbf{p}_0 \cdot 1 \cdot \frac{\mathbf{p}_0}{2c} = \frac{\mathbf{p}_0^2}{2c}. \tag{C.112}$$

Now we analyze the distribution of $\mathbf{p}_0$. Since $\mathbf{z}$ follows a spherical Gaussian distribution $\mathcal{N}(0, \gamma \boldsymbol{I}_c)$, we know that the entries of $\mathbf{z}$ are totally independent. Besides, for each entry $\mathbf{z}_i (i \in [c])$, we have $|\mathbf{z}_i| < \gamma$ with probability ξ, where $\xi \approx 0.68$ is an absolute constant. Therefore, with probability ξ^c, forall entries $\mathbf{z}_i (i \in [c])$, we have $|\mathbf{z}_i| < \gamma$. In this case,

$$\mathbf{p}_0 = \frac{\exp\big(\min_{i \in [c]} \mathbf{z}_i\big)}{\sum_{i=1}^{c} \exp(\mathbf{z}_i)} \geqslant \frac{\exp(-\gamma)}{c \exp(\gamma)}. \tag{C.113}$$

In other cases, we know that $\mathbf{p}_0 > 0$. Thus,

$$\lambda_{c-1} = \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(0, \gamma \boldsymbol{I}_c)} \left[\min_{\boldsymbol{v} \in S, \|v\|=1} \boldsymbol{v}^\top \mathbf{A} \boldsymbol{v} \right] \geqslant \xi^c \cdot \frac{\left(\frac{\exp(-\gamma)}{c \exp(\gamma)} \right)^2}{2c}. \tag{C.114}$$

The right hand side is independent of n. Therefore, $\lambda_{c-1} > 0$, which means that $\widetilde{\mathbf{A}}$ has exactly $(c-1)$ positive eigenvalues and a 0 eigenvalue, and the eigenvalue gap between the smallest positive eigenvalue and 0 is independent of n.

Hence we complete the proof.

□

Projecting Hessians onto Finite Dimensions

In this section we will develop some technical tools for analyzing the eigenvalues and eigenvectors of the output Hessians and the full layer-wise Hessians. In particular, we will project both infinite dimensional matrices to $c \times c$ matrices.

First, we prove a technical lemma that will be very useful when we bound the Frobenius norm of the difference between infinite size matrices.

Lemma C.2.16. *Let $p(\mathbf{A}, \mathbf{D}, \mathbf{x})$ be a homogeneous polynomial of $\mathbf{A}$, $\mathbf{D}$, and $\mathbf{x}$ and is degree 1 in $\mathbf{A}$, degree 2 in $\mathbf{D}$, and degree 2 in $\mathbf{x}$. Suppose the coefficients in p are upper bounded in ℓ_1-norm by an absolute constant μ. Also let $\mathbf{D}'$ be an independent copy of $\mathbf{D}$ and $\mathbf{x}''$ be an independent copy of $\mathbf{x}$ independent to $\mathbf{D}$ and $\mathbf{A}$. Then with probability 1 over $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$, we have*

$$\lim_{n\to\infty} \mathbb{E}\left[p(\mathbf{A}, \mathbf{D}, \mathbf{x})\right] = \mathbb{E}\left[p(\mathbf{A}, \mathbf{D}', \mathbf{x}'')\right] \tag{C.115}$$

Proof of Lemma C.2.16. Fix any $\epsilon > 0$. Assume that the homogeneous polynomial is of the form

$$p(\mathbf{A}, \mathbf{D}, \mathbf{x}) = \sum_{i=1}^{m} c_i \mathbf{A}_{s(i),t(i)} \mathbf{D}_{u(i),u(i)} \mathbf{D}_{v(i),v(i)} \mathbf{x}_{p(i)} \mathbf{x}_{q(i)}, \tag{C.116}$$

for coefficients c_i, then from linearity of expectation we know

$$\mathbb{E}[p(\mathbf{A}, \mathbf{D}, \mathbf{x})] = \sum_{i=1}^{m} c_i \mathbb{E}[\mathbf{A}_{s(i),t(i)} \mathbf{D}_{u(i),u(i)} \mathbf{D}_{v(i),v(i)} \mathbf{x}_{p(i)} \mathbf{x}_{q(i)}]. \tag{C.117}$$

Hence

$$
\begin{aligned}
&|\mathbb{E}\left[p(\mathbf{A},\mathbf{D},\mathbf{x})\right]-\mathbb{E}\left[p(\mathbf{A},\mathbf{D}',\mathbf{x}'')\right]| \\
\leqslant&\sum_{i=1}^{m} c_i|\mathbb{E}[\mathbf{A}_{s(i),t(i)}\mathbf{D}_{u(i),u(i)}\mathbf{D}_{v(i),v(i)}\mathbf{x}_{p(i)}\mathbf{x}_{q(i)}]-\mathbb{E}[\mathbf{A}_{s(i),t(i)}\mathbf{D}_{u(i),u(i)}\mathbf{D}_{v(i),v(i)}\mathbf{x}_{p(i)}\mathbf{x}_{q(i)}]|
\end{aligned}
\tag{C.118}
$$

Since the entries of $\mathbf{D}$ can only be 0 or 1, we have

$$
\begin{aligned}
&\mathbb{E}[\mathbf{A}_{s(i),t(i)}\mathbf{D}_{u(i),u(i)}\mathbf{D}_{v(i),v(i)}\mathbf{x}_{p(i)}\mathbf{x}_{q(i)}] \\
=&\Pr\left[\mathbf{D}_{u(i),u(i)}=\mathbf{D}_{v(i),v(i)}=1\right]\mathbb{E}[\mathbf{A}_{s(i),t(i)}\mathbf{x}_{p(i)}\mathbf{x}_{q(i)}|\mathbf{D}_{u(i),u(i)}=\mathbf{D}_{v(i),v(i)}=1] \\
=&\frac{1}{4}\mathbb{E}[\mathbf{A}_{s(i),t(i)}\mathbf{x}_{p(i)}\mathbf{x}_{q(i)}|\mathbf{D}_{u(i),u(i)}=\mathbf{D}_{v(i),v(i)}=1].
\end{aligned}
\tag{C.119}
$$

The last equality holds since $\mathbf{u}$ converges in distribution to a spherical Gaussian, and its entry-wise activations $\mathbf{D}$ follows a $p=\frac{1}{2}$ Bernoulli distribution. Assume $\sum_{i=1}^{m}|c_i|\geqslant\mu$, that the ℓ_1 norm of the coefficients is upper bounded by some constant μ. Set $\epsilon'=\frac{\epsilon}{\mu}$. To prove this lemma it is sufficient to prove that each term of the polynomial are sufficiently small, namely, for any index,

$$
\begin{aligned}
&|\mathbb{E}[\mathbf{A}_{s(i),t(i)}\mathbf{x}_{p(i)}\mathbf{x}_{q(i)}|\mathbf{D}_{u(i),u(i)}=\mathbf{D}_{v(i),v(i)}=1] \\
&\quad-\mathbb{E}[\mathbf{A}_{s(i),t(i)}\mathbf{x}''_{p(i)}\mathbf{x}''_{q(i)}|\mathbf{D}'_{u(i),u(i)}=\mathbf{D}'_{v(i),v(i)}=1]| \\
&|\mathbb{E}[\mathbf{A}_{s(i),t(i)}\mathbf{x}_{p(i)}\mathbf{x}_{q(i)}|\mathbf{D}_{u(i),u(i)}=\mathbf{D}_{v(i),v(i)}=1]-\mathbb{E}[\mathbf{A}_{s(i),t(i)}\mathbf{x}''_{p(i)}\mathbf{x}''_{q(i)}]|<\epsilon'.
\end{aligned}
\tag{C.120}
$$

Fix a set of index s,t,p,q,u,v, for simplicity of notation, we use the abbreviation $\mathbb{E}\left[\mathbf{A}_{st}\mathbf{x}_p\mathbf{x}_q|_{\mathbf{D}}\right]$ to denote $\mathbb{E}[\mathbf{A}_{s(i),t(i)}\mathbf{x}_{p(i)}\mathbf{x}_{q(i)}|\mathbf{D}_{u(i),u(i)}=\mathbf{D}_{v(i),v(i)}=1]$. Since $\mathbf{x}$ is of rectified Gaussian with the covariance of the initial Gaussian distribution being the identity, $\mathbf{x}_p$ and $\mathbf{x}_q$ shares the same density function when $x>0$, namely $f(x)=$

$\frac{1}{\sqrt{2\pi}}\exp(-x^2/2)$. Note that

$$\iint\limits_{\mathbb{R}^+\times\mathbb{R}^+} xy\ f(x)f(y)\ dx\ dy = \mathbb{E}\left[\mathbf{x}_i\mathbf{x}_j\right] = \mathbb{E}\left[\mathbf{x}_i\right]\mathbb{E}\left[\mathbf{x}_j\right] = \frac{1}{2\pi}. \tag{C.121}$$

Fix some $\beta < \frac{\alpha}{2}$, we have

$$\begin{aligned}
&\left|\mathbb{E}\left[\mathbf{A}_{st}\mathbf{x}_p\mathbf{x}_q|_{\mathbf{D}}\right] - \mathbb{E}\left[\mathbf{A}_{st}\mathbf{x}''_p\mathbf{x}''_q\right]\right| \\
&= \left| \iint\limits_{\mathbb{R}^+\times\mathbb{R}^+} \mathbb{E}\left[\mathbf{A}_{st}|_{\mathbf{D},\mathbf{x}_p=x,\mathbf{x}_q=y}\right] xy\ f(x)f(y)\ dx\ dy - \iint\limits_{\mathbb{R}^+\times\mathbb{R}^+} \mathbb{E}\left[\mathbf{A}_{st}\right] xy\ f(x)f(y)\ dx\ dy \right| \\
&\leqslant \iint\limits_{\mathbb{R}^+\times\mathbb{R}^+} \left|\mathbb{E}\left[\mathbf{A}_{st}|_{\mathbf{D},\mathbf{x}_p=x,\mathbf{x}_q=y}\right] - \mathbb{E}\left[\mathbf{A}_{st}\right]\right| xy\ f(x)f(y)\ dx\ dy \\
&= \iint\limits_{[0,n^\beta]\times[0,n^\beta]} \left|\mathbb{E}\left[\mathbf{A}_{st}|_{\mathbf{D},\mathbf{x}_p=x,\mathbf{x}_q=y}\right] - \mathbb{E}\left[\mathbf{A}_{st}\right]\right| xy\ f(x)f(y)\ dx\ dy \\
&\quad + \iint\limits_{\mathbb{R}^+\times\mathbb{R}^+\setminus([0,n^\beta]\times[0,n^\beta])} \left|\mathbb{E}\left[\mathbf{A}_{st}|_{\mathbf{D},\mathbf{x}_p=x,\mathbf{x}_q=y}\right] - \mathbb{E}\left[\mathbf{A}_{st}\right]\right| xy\ f(x)f(y)\ dx\ dy.
\end{aligned} \tag{C.122}$$

From Corollary C.2.14 we have, for any indices s, t, for sufficiently large n, for any $(x, y) \in [0, n^\beta] \times [0, n^\beta]$,

$$\left|\mathbb{E}\left[\mathbf{A}_{st}|_{\mathbf{D},\mathbf{x}_p=x,\mathbf{x}_q=y}\right] - \mathbb{E}\left[\mathbf{A}_{st}\right]\right| < \epsilon' \tag{C.123}$$

Thus

$$\begin{aligned}
&\iint\limits_{[0,n^\beta]\times[0,n^\beta]} \left|\mathbb{E}\left[\mathbf{A}_{st}|_{\mathbf{D},\mathbf{x}_p=x,\mathbf{x}_q=y}\right] - \mathbb{E}\left[\mathbf{A}_{st}\right]\right| xy\ f(x)f(y)\ dx\ dy \\
&\leqslant \epsilon' \iint\limits_{[0,n^\beta]\times[0,n^\beta]} xy\ f(x)f(y)\ dx\ dy = \frac{\epsilon'}{2\pi}.
\end{aligned} \tag{C.124}$$

Now we consider the other integral. First note that since $\mathbf{A}_{st}$ is either $\mathbf{p}_i - \mathbf{p}_i^2$ or $-\mathbf{p}_i\mathbf{p}_j$ for some i, j, and $\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_i + \mathbf{p}_j \in (0,1)$ as it is the output of the softmax function, we have $\mathbf{A}_{st} \in (-\frac{1}{4}, \frac{1}{4})$. It follows that $|\mathbb{E}\left[\mathbf{A}_{st}|_{\mathbf{D},\mathbf{x}_p=x,\mathbf{x}_q=y}\right] - \mathbb{E}\left[\mathbf{A}_{st}\right]| \leqslant \frac{1}{2}$. Therefore

$$\begin{aligned}
&\iint\limits_{\mathbb{R}^+\times\mathbb{R}^+\setminus([0,n^\beta]\times[0,n^\beta])} |\mathbb{E}\left[\mathbf{A}_{st}|_{\mathbf{D},\mathbf{x}_p=x,\mathbf{x}_q=y}\right] - \mathbb{E}\left[\mathbf{A}_{st}\right]| xy\ f(x)f(y)\ dx\ dy \\
&\leqslant \frac{1}{2} \iint\limits_{\mathbb{R}^+\times\mathbb{R}^+\setminus([0,n^\beta]\times[0,n^\beta])} xy\ \frac{e^{-x^2/2}}{\sqrt{2\pi}}\frac{e^{-y^2/2}}{\sqrt{2\pi}}\ dx\ dy \\
&\leqslant \frac{1}{2}\cdot\frac{1}{2\pi}\int_{n^\beta}^{\infty} e^{-x^2/2}x\ dx \int_{\mathbb{R}^+} e^{-y^2/2}y\ dy + \frac{1}{2}\cdot\frac{1}{2\pi}\int_{\mathbb{R}^+} e^{-x^2/2}x\ dx \int_{n^\beta}^{\infty} e^{-y^2/2}y\ dy \\
&= \frac{1}{2\pi}e^{-n^{2\beta}},
\end{aligned} \tag{C.125}$$

which decreases below $\epsilon'/2$ for sufficiently large n. As both terms in Equation C.122 are less than $\epsilon'/2$ as $n \to \infty$, we have $|\mathbb{E}\left[\mathbf{A}_{st}\mathbf{x}_p\mathbf{x}_q|_{\mathbf{D}}\right] - \mathbb{E}\left[\mathbf{A}_{st}\mathbf{x}''_p\mathbf{x}''_q\right]| < \epsilon'$. Which completes the proof of this lemma. □

We then generalize this lemma for a degree ten homogeneous polynomial, in which the monomials are roughly multiplied with an independent copy of itself (except for $\mathbf{A}$).

Corollary C.2.17. *Let $p(\mathbf{A}, \mathbf{D}, \mathbf{x}, \bar{\mathbf{A}}, \bar{\mathbf{D}}, \bar{\mathbf{x}})$ be a homogeneous polynomial of $\mathbf{A}$, $\mathbf{D}$, $\mathbf{x}$, $\bar{\mathbf{A}}$, $\bar{\mathbf{D}}$, and $\bar{\mathbf{x}}$. Let it be degree 1 in $\mathbf{A}$, $\bar{\mathbf{A}}$, degree 2 in $\mathbf{D}$, $\bar{\mathbf{D}}$, and degree 2 in $\mathbf{x}$,$\bar{\mathbf{x}}$. Suppose the coefficients in p are upper bounded in ℓ_1-norm by an absolute constant μ. Also let $\mathbf{D}'$ be an independent copy of $\mathbf{D}$ and $\mathbf{x}''$ be an independent copy of $\mathbf{x}$ independent to $\mathbf{D}$ and $\mathbf{A}$. Moreover, let $(\bar{\mathbf{A}}, \bar{\mathbf{D}}, \bar{\mathbf{x}}, \bar{\mathbf{D}}', \bar{\mathbf{x}}'')$ be an independent copy of*

$(\mathbf{A}, \mathbf{D}, \mathbf{x}, \mathbf{D}', \mathbf{x}'')$. *Then with probability 1 over* $\boldsymbol{W}^{(1)}$ *and* $\boldsymbol{W}^{(2)}$*, we have*

$$\lim_{n\to\infty} \mathbb{E}\left[p(\mathbf{A}, \mathbf{D}, \mathbf{x}, \bar{\mathbf{A}}, \bar{\mathbf{D}}, \bar{\mathbf{x}})\right] = \mathbb{E}\left[p(\mathbf{A}, \mathbf{D}', \mathbf{x}'', \bar{\mathbf{A}}, \bar{\mathbf{D}}', \bar{\mathbf{x}}'')\right]. \tag{C.126}$$

Proof of Corollary C.2.17. For simplicity of notations, we here denote $\mathrm{s}_{ijuvrs} = \mathbf{A}_{ij}\mathbf{D}_{vv}\mathbf{D}_{ww}\mathbf{x}_r\mathbf{x}_s$, $\mathrm{s}'_{ijuvrs} = \mathbf{A}_{ij}\mathbf{D}'_{vv}\mathbf{D}'_{ww}\mathbf{x}''_r\mathbf{x}''_s$. Similarly, we also denote $\mathrm{t}_{klpqtu} = \bar{\mathbf{A}}_{kl}\bar{\mathbf{D}}_{pp}\bar{\mathbf{D}}_{qq}\bar{\mathbf{x}}_t\bar{\mathbf{x}}_u$ and $\mathrm{t}'_{klpqtu} = \bar{\mathbf{A}}_{kl}\bar{\mathbf{D}}'_{pp}\bar{\mathbf{D}}'_{qq}\bar{\mathbf{x}}''_t\bar{\mathbf{x}}''_u$. As there is no confusion on indexing, we will also omit the subscripts and use s, t.

Fix any $\epsilon > 0$, Following the argument of the proof of Lemma C.2.16, it is sufficient to prove this corollary by showing for any indexing,

$$\begin{aligned}&|\mathbb{E}[\mathbf{A}_{ij}\bar{\mathbf{A}}_{kl}\mathbf{D}_{vv}\mathbf{D}_{ww}\bar{\mathbf{D}}_{pp}\bar{\mathbf{D}}_{qq}\mathbf{x}_r\mathbf{x}_s\bar{\mathbf{x}}_t\bar{\mathbf{x}}_u] - \mathbb{E}[\mathbf{A}_{ij}\bar{\mathbf{A}}_{kl}\mathbf{D}'_{vv}\mathbf{D}'_{ww}\bar{\mathbf{D}}'_{pp}\bar{\mathbf{D}}'_{qq}\mathbf{x}''_r\mathbf{x}''_s\bar{\mathbf{x}}''_t\bar{\mathbf{x}}''_u]| \\ &= |\mathbb{E}[\mathrm{st}] - \mathbb{E}[\mathrm{s}'\mathrm{t}']| < \frac{\epsilon}{\mu}.\end{aligned} \tag{C.127}$$

First note that since $|\mathbf{A}_{ij}| < \frac{1}{4}$ and $|\mathbf{D}_{ii}| \leqslant 1$ for all i, j, we have

$$|\mathbb{E}[\mathrm{s}]| = |\mathbb{E}[\mathbf{A}_{ij}\mathbf{D}_{vv}\mathbf{D}_{ww}\mathbf{x}_r\mathbf{x}_s]| \leqslant \frac{1}{4}|\mathbb{E}[\mathbf{x}_r\mathbf{x}_s]| = \frac{1}{8\pi}. \tag{C.128}$$

The same argument also applies to s', t, and t'. Also, by Lemma C.2.16, for sufficiently large n we have $|\mathbb{E}[\mathrm{s}] - \mathbb{E}[\mathrm{s}']| < \epsilon'$ and $|\mathbb{E}[\mathrm{t}] - \mathbb{E}[\mathrm{t}']| < \epsilon'$. Since by construction s and t are independent, we have

$$\begin{aligned}|\mathbb{E}[\mathrm{st}] - \mathbb{E}[\mathrm{s}'\mathrm{t}']| &= |\mathbb{E}[\mathrm{s}]\mathbb{E}[\mathrm{t}] - \mathbb{E}[\mathrm{s}']\mathbb{E}[\mathrm{t}']| \\ &= |\mathbb{E}[\mathrm{s}]\mathbb{E}[\mathrm{t}] - \mathbb{E}[\mathrm{s}]\mathbb{E}[\mathrm{t}'] + \mathbb{E}[\mathrm{s}]\mathbb{E}[\mathrm{t}'] - \mathbb{E}[\mathrm{s}']\mathbb{E}[\mathrm{t}']| \\ &\leqslant |\mathbb{E}[\mathrm{s}]||\mathbb{E}[\mathrm{t}] - \mathbb{E}[\mathrm{t}']| + |\mathbb{E}[\mathrm{t}']||\mathbb{E}[\mathrm{s}] - \mathbb{E}[\mathrm{s}']| \\ &\leqslant \frac{1}{8\pi}\epsilon' + \frac{1}{8\pi}\epsilon' < \epsilon',\end{aligned} \tag{C.129}$$

which completes the proof of Corollary C.2.17. □

Now we formally begin our analysis. We start from $\boldsymbol{M}^{(1)} = \mathbb{E}\left[\mathbf{D}\boldsymbol{W}^{(2)\mathsf{T}}\mathbf{A}\boldsymbol{W}^{(2)}\mathbf{D}\right]$, the output Hessian of the first layer. The output Hessian of the second layer is just $\mathbb{E}\left[\mathbf{A}\right]$, which had been analyzed in Section C.2.2. In this section we will neglect the superscript for $\boldsymbol{M}^{(1)}$ and use $\boldsymbol{M}$ as there is no confusion. Also, we use $\boldsymbol{W}$ to denote $\boldsymbol{W}^{(2)}$ unless specified otherwise. We first state our main lemma of projecting $\boldsymbol{M}$.

Lemma C.2.18. *With probability 1 over $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$,*

$$\lim_{n\to\infty} \frac{\left\|\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^{\mathsf{T}}\right\|_F^2}{\|\boldsymbol{M}\|_F^2} = 1. \tag{C.130}$$

Proof of Lemma C.2.18. To prove the equivalence between $\left\|\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^{\mathsf{T}}\right\|_F^2$ and $\|\boldsymbol{M}\|_F^2$, we need to introduce a bridging term

$$\boldsymbol{M}^* \triangleq \mathbb{E}[\mathbf{D}'\boldsymbol{W}^{(2)\mathsf{T}}\mathbf{A}\boldsymbol{W}^{(2)}\mathbf{D}'] \tag{C.131}$$

where $\mathbf{D}'$ is an independent copy of $\mathbf{D}$ and also independent of $\mathbf{A}$. Essentially $\boldsymbol{M}^*$ is the matrix which has the same expression as $\boldsymbol{M}$ except that we assume $\mathbf{D}$ is independent of $\mathbf{A}$ in $\boldsymbol{M}^*$. Informally, the proof strategy of Lemma C.2.18 is

$$\left\|\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^{\mathsf{T}}\right\|_F^2 \approx \left\|\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^{\mathsf{T}}\right\|_F^2 \approx \|\boldsymbol{M}^*\|_F^2 \approx \|\boldsymbol{M}\|_F^2. \tag{C.132}$$

We now formally establish this equivalence.

Then we look into the structures of the bridging matrix $\boldsymbol{M}^*$. It is simple to analyze as we assumed the independence between $\mathbf{A}$ and $\mathbf{D}'$. Formally,

Lemma C.2.19. *With probability 1 over $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$,*

$$\boldsymbol{M}^* = \frac{1}{4}\left(\boldsymbol{W}^{\mathsf{T}}\mathbb{E}[\mathbf{A}]\boldsymbol{W} + \operatorname{diag}(\boldsymbol{W}^{\mathsf{T}}\mathbb{E}[\mathbf{A}]\boldsymbol{W})\right). \tag{C.133}$$

Moreover, $\|\boldsymbol{M}^*\|$ and $\|\boldsymbol{M}^*\|_F^2$ are bounded below by some nonzero constant and bounded above by some constant.

Proof of Lemma C.2.19. First note that since $\mathbf{D}'$ is the activation of $\mathbf{u}'$, which converges to a spherical Gaussian with probability 1 over $\boldsymbol{W}^{(1)}$ and is independent with $\mathbf{A}$, each diagonal entry of $\mathbf{D}$ is a Bernoulli random variable with $p = \frac{1}{2}$. For $i, j \in [n]$, when $i \neq j$, we have

$$\begin{aligned}\boldsymbol{M}^*_{ij} &= \mathbb{E}[\mathbf{D}'_{ii}(\boldsymbol{W}^\top \mathbf{A} \boldsymbol{W})_{ij} \mathbf{D}'_{jj}] \\ &= \mathbb{E}[\mathbf{D}'_{ii}]\mathbb{E}[\mathbf{D}'_{jj}]\mathbb{E}[(\boldsymbol{W}^\top \mathbf{A} \boldsymbol{W})_{ij}] \\ &= \frac{1}{4}(\boldsymbol{W}^\top \mathbb{E}[\mathbf{A}] \boldsymbol{W})_{ij}.\end{aligned} \tag{C.134}$$

When $i = j$,

$$\begin{aligned}\boldsymbol{M}^*_{i,i} &= \mathbb{E}[\mathbf{D}'_{ii}(\boldsymbol{W}^\top \mathbf{A} \boldsymbol{W})_{ii} \mathbf{D}'_{ii}] \\ &= \mathbb{E}[\mathbf{D}'_{ii}]\mathbb{E}[(\boldsymbol{W}^\top \mathbf{A} \boldsymbol{W})_{ii}] \\ &= \frac{1}{2}(\boldsymbol{W}^\top \mathbb{E}[\mathbf{A}] \boldsymbol{W})_{i,j}.\end{aligned} \tag{C.135}$$

Thus

$$\boldsymbol{M}^* = \frac{1}{4}\left(\boldsymbol{W}^\top \mathbb{E}[\mathbf{A}] \boldsymbol{W} + \operatorname{diag}(\boldsymbol{W}^\top \mathbb{E}[\mathbf{A}] \boldsymbol{W})\right). \tag{C.136}$$

Now we show the lower bound and upper bound on norms of $\boldsymbol{M}^*$.

Since $\langle \mathbb{E}[\boldsymbol{W}^\top \mathbf{A} \boldsymbol{W}], \operatorname{diag}(\mathbb{E}[\boldsymbol{W}^\top \mathbf{A} \boldsymbol{W}]) \rangle \geqslant 0$, we have

$$\|\boldsymbol{M}^*\|_F \geqslant \left\|\mathbb{E}[\boldsymbol{W}^\top \mathbf{A} \boldsymbol{W}]\right\|_F = \left\|\boldsymbol{W}^\top \widetilde{\mathbf{A}} \boldsymbol{W}\right\|_F. \tag{C.137}$$

Since $\boldsymbol{W}\boldsymbol{W}^\top$ converges to $\boldsymbol{I}_c$ in spectral norm from Lemma C.2.5, we have for sufficiently large n, the smallest singular value of $\boldsymbol{W}$ is larger than $\frac{1}{2}$. Moreover, since $\mathbb{E}[\mathbf{A}]$ admits an eigenvalue that is bounded below by some constants $\eta \triangleq$

$\xi^c \cdot \left(\frac{\exp(-\gamma)}{c\exp(\gamma)}\right)^2/2c$ where $\xi \approx 0.68$ is an absolute constant and $\gamma = \frac{(\pi-1)^2}{4\pi^2}$ as shown in Lemma C.2.15, there exists an eigenvalue of $\boldsymbol{M}^* = \boldsymbol{W}^\mathsf{T}\mathbb{E}[\mathbf{A}]\boldsymbol{W}$ that is larger than $\frac{\eta}{4}$. Hence for large n, $\|\boldsymbol{M}^*\|$ is bounded from below by $\frac{\eta}{4}$, and hence $\|\boldsymbol{M}^*\|_F^2$.

Besides, since $\mathbf{D}$ is a diagonal matrix with 0/1 entries, and the absolute value of each entry of $\mathbf{A}$ is bounded by 1, we have

$$\|\boldsymbol{M}\|_F = \left\|\mathbb{E}[\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}]\right\|_F \leqslant \left\|\mathbb{E}[\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}]\right\|_F \leqslant \|\boldsymbol{W}\|_F^2\|\mathbf{A}\|_F \leqslant c\|\boldsymbol{W}\|_F^2. \tag{C.138}$$

From Lemma C.2.3, we know that with probability 1, $\|\boldsymbol{W}\|_F^2 \leqslant 2c$, therefore, $\|\boldsymbol{M}\|_F$ is upper bounded by $2c^2$, which is independent of n. □

Lemma C.2.20. *With probability 1 over $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$,*

$$\lim_{n\to\infty} \frac{\|\boldsymbol{M}\|_F^2}{\|\boldsymbol{M}^*\|_F^2} = 1.$$

Proof of Lemma C.2.20.

Recall that $\boldsymbol{M}^* \triangleq \mathbb{E}[\mathbf{D}'\boldsymbol{W}^{(2)\mathsf{T}}\mathbf{A}\boldsymbol{W}^{(2)}\mathbf{D}']$ where $\mathbf{D}'$ is an independent copy of $\mathbf{D}$ and also independent of $\mathbf{A}$. Since we will only explicitly use $\boldsymbol{W}^{(2)}$ in this proof, for simplicity of notation, we will omit its superscript and use $\boldsymbol{W}$. Let $(\bar{\mathbf{D}}, \bar{\mathbf{A}})$ be an independent copy of $(\mathbf{D}, \mathbf{A})$, then

$$\begin{aligned}
\|\boldsymbol{M}\|_F^2 &= \left\|\mathbb{E}[\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}]\right\|_F^2 \\
&= \mathbb{E}\left[\langle \mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}, \bar{\mathbf{D}}\boldsymbol{W}^\mathsf{T}\bar{\mathbf{A}}\boldsymbol{W}\bar{\mathbf{D}}\rangle\right] \\
&= \mathbb{E}\left[\mathrm{tr}\left(\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}\bar{\mathbf{D}}\boldsymbol{W}^\mathsf{T}\bar{\mathbf{A}}\boldsymbol{W}\bar{\mathbf{D}}\right)\right] \\
&= \mathbb{E}\left[\mathrm{tr}\left(\boldsymbol{W}\bar{\mathbf{D}}\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}\bar{\mathbf{D}}\boldsymbol{W}^\mathsf{T}\bar{\mathbf{A}}\right)\right].
\end{aligned} \tag{C.139}$$

Expressing the term inside the expectation as a polynomial of entries of $\mathbf{A}$, $\mathbf{D}$, $\bar{\mathbf{A}}$

and $\bar{\mathbf{D}}$, we get

$$\begin{aligned}
&\operatorname{tr}\left(\boldsymbol{W}\bar{\mathbf{D}}\mathbf{D}\boldsymbol{W}^{\mathsf{T}}\mathbf{A}\boldsymbol{W}\mathbf{D}\bar{\mathbf{D}}\boldsymbol{W}^{\mathsf{T}}\bar{\mathbf{A}}\right)\\
=&\sum_{i=1}^{c}\left(\boldsymbol{W}\bar{\mathbf{D}}\mathbf{D}\boldsymbol{W}^{\mathsf{T}}\mathbf{A}\boldsymbol{W}\mathbf{D}\bar{\mathbf{D}}\boldsymbol{W}^{\mathsf{T}}\bar{\mathbf{A}}\right)_{i,i}\\
=&\sum_{i,j=1}^{c}\left(\boldsymbol{W}\bar{\mathbf{D}}\mathbf{D}\boldsymbol{W}^{\mathsf{T}}\mathbf{A}\right)_{i,j}\left(\boldsymbol{W}\mathbf{D}\bar{\mathbf{D}}\boldsymbol{W}^{\mathsf{T}}\bar{\mathbf{A}}\right)_{j,i}\\
=&\sum_{i,j=1}^{c}\left(\sum_{k=1}^{c}\sum_{l=1}^{n}\boldsymbol{W}_{i,l}\boldsymbol{W}_{k,l}\mathbf{D}_{l,l}\mathbf{D}_{l,l}\mathbf{A}_{k,j}\right)\left(\sum_{s=1}^{c}\sum_{t=1}^{n}\boldsymbol{W}_{j,t}\boldsymbol{W}_{s,t}\bar{\mathbf{D}}_{t,t}\bar{\mathbf{D}}_{t,t}\mathbf{A}_{s,i}\right)\\
=&\sum_{i,j,k,s=1}^{c}\sum_{l,t=1}^{n}\boldsymbol{W}_{i,l}\boldsymbol{W}_{k,l}\boldsymbol{W}_{j,t}\boldsymbol{W}_{s,t}\bar{\mathbf{A}}_{k,j}\mathbf{A}_{s,i}\bar{\mathbf{D}}_{l,l}\mathbf{D}_{l,l}\bar{\mathbf{D}}_{t,t}\mathbf{D}_{t,t}.
\end{aligned}\tag{C.140}$$

The monomials are $\bar{\mathbf{A}}_{k,j}\mathbf{A}_{s,i}\bar{\mathbf{D}}_{l,l}\mathbf{D}_{l,l}\bar{\mathbf{D}}_{t,t}\mathbf{D}_{t,t}$, and the corresponding coefficients are $\boldsymbol{W}_{i,l}\boldsymbol{W}_{k,l}\boldsymbol{W}_{j,t}\boldsymbol{W}_{s,t}$. Now we can bound the ℓ_1 norm of the coefficient of this polynomial as follows:

$$\begin{aligned}
&\left\|\sum_{i,j,k,s=1}^{c}\sum_{l,t=1}^{n}\boldsymbol{W}_{i,l}\boldsymbol{W}_{k,l}\boldsymbol{W}_{j,t}\boldsymbol{W}_{s,t}\right\|_1\\
\leqslant&\sum_{i,j,k,s=1}^{c}\sum_{l,t=1}^{n}|\boldsymbol{W}_{i,l}|\cdot|\boldsymbol{W}_{k,l}|\cdot|\boldsymbol{W}_{j,t}|\cdot|\boldsymbol{W}_{s,t}|\\
=&\left(\sum_{i,k=1}^{c}\sum_{l=1}^{n}|\boldsymbol{W}_{i,l}|\cdot|\boldsymbol{W}_{k,l}|\right)\left(\sum_{j,s=1}^{c}\sum_{t=1}^{n}|\boldsymbol{W}_{j,t}|\cdot|\boldsymbol{W}_{s,t}|\right)\\
\leqslant&\left(\sum_{i,k=1}^{c}\sum_{l=1}^{n}\frac{\boldsymbol{W}_{i,l}^2+\boldsymbol{W}_{k,l}^2}{2}\right)\left(\sum_{j,s=1}^{c}\sum_{t=1}^{n}\frac{\boldsymbol{W}_{j,t}^2+\boldsymbol{W}_{s,t}^2}{2}\right)\\
=&\left(\sum_{i,k=1}^{c}\frac{\|\boldsymbol{W}_i\|^2+\|\boldsymbol{W}_k\|^2}{2}\right)\left(\sum_{j,s=1}^{c}\frac{\|\boldsymbol{W}_j\|^2+\|\boldsymbol{W}_s\|^2}{2}\right)\\
=&(c\|\boldsymbol{W}\|_F^2)^2=c^2\|\boldsymbol{W}\|_F^4.
\end{aligned}\tag{C.141}$$

From Lemma C.2.3 we know that $\|\boldsymbol{W}\|_F^2 = O(c)$ with probability 1 over $\boldsymbol{W}$, so the coefficient of this polynomial is ℓ_1-norm bounded.

For any $\epsilon > 0$, fix ϵ. Note that $\|\boldsymbol{M}^*\|_F^2$ is just substituting $\mathbf{D}, \bar{\mathbf{D}}$ by $\mathbf{D}', \bar{\mathbf{D}}'$ in the polynomial characterized by Equation C.140. From Corollary C.2.17 we have the convergence of the difference of the expectation of the two polynomials, namely $|\|\boldsymbol{M}\|_F^2 - \|\boldsymbol{M}^*\|_F^2| < \epsilon$ for sufficiently large n. Since the spectral norm of $\boldsymbol{M}^*$ is on the order of constant from Lemma C.2.19, we have $\lim_{n\to\infty} \|\boldsymbol{M}\|_F^2 / \|\boldsymbol{M}^*\|_F^2 = 1$. $\square$

Lemma C.2.21. *For all $i, j \in [c]$, $\lim_{n\to\infty}((\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\mathsf{T})_{i,j} - (\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\mathsf{T})_{i,j}) = 0$. Thus,*

$$\lim_{n\to\infty} \frac{\left\|\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\mathsf{T}\right\|_F^2}{\|\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\mathsf{T}\|_F^2} = 1.$$

Proof of Lemma C.2.21. This proof is very similar to that of Lemma C.2.20. First, we focus on a single entry of the matrix $\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\mathsf{T}$ and express it as a polynomial of entries of $\mathbf{A}$ and $\mathbf{D}$:

$$\begin{aligned}
(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\mathsf{T})_{i,j} &= \mathbb{E}[(\boldsymbol{W}\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}\boldsymbol{W}^\mathsf{T})_{i,j}] \\
&= \mathbb{E}\left[\sum_{k=1}^{c} (\boldsymbol{W}\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A})_{i,k} (\boldsymbol{W}\mathbf{D}\boldsymbol{W}^\mathsf{T})_{k,j}\right] \\
&= \mathbb{E}\left[\sum_{k=1}^{c} \left(\sum_{s=1}^{c}\sum_{l=1}^{n} \boldsymbol{W}_{i,l}\boldsymbol{W}_{s,l}\mathbf{D}_{l,l}\mathbf{A}_{s,k}\right)\left(\sum_{t=1}^{n} \boldsymbol{W}_{k,j}\boldsymbol{W}_{j,t}\mathbf{D}_{t,t}\right)\right] \\
&= \mathbb{E}\left[\sum_{k,s=1}^{c}\sum_{l,t=1}^{n} \boldsymbol{W}_{i,l}\boldsymbol{W}_{s,l}\boldsymbol{W}_{k,t}\boldsymbol{W}_{j,t}\mathbf{A}_{s,k}\mathbf{D}_{l,l}\mathbf{D}_{t,t}\right].
\end{aligned} \tag{C.142}$$

Then we bound the ℓ_1 norm of the coefficients of this polynomial as follows:

$$\begin{aligned}
&\left\| \sum_{k,s=1}^{c} \sum_{l,t=1}^{n} \boldsymbol{W}_{i,l} \boldsymbol{W}_{s,l} \boldsymbol{W}_{k,t} \boldsymbol{W}_{j,t} \right\|_1 \\
\leqslant & \sum_{k,s=1}^{c} \sum_{l,t=1}^{n} |\boldsymbol{W}_{i,l}| \cdot |\boldsymbol{W}_{s,l}| \cdot |\boldsymbol{W}_{k,t}| \cdot |\boldsymbol{W}_{j,t}| \\
= & \left(\sum_{s=1}^{c} \sum_{l=1}^{n} |\boldsymbol{W}_{i,l}| \cdot |\boldsymbol{W}_{s,l}| \right) \left(\sum_{k=1}^{c} \sum_{t=1}^{n} |\boldsymbol{W}_{k,t}| \cdot |\boldsymbol{W}_{j,t}| \right) \\
\leqslant & \left(\sum_{s=1}^{c} \sum_{l=1}^{n} \frac{\boldsymbol{W}_{i,l}^2 + \boldsymbol{W}_{s,l}^2}{2} \right) \left(\sum_{k=1}^{c} \sum_{t=1}^{n} \frac{\boldsymbol{W}_{k,t}^2 + \boldsymbol{W}_{j,t}^2}{2} \right) \\
= & \left(c \|\boldsymbol{W}_i\|^2 + \|\boldsymbol{W}\|_F^2 \right) \left(c \|\boldsymbol{W}_j\|^2 + \|\boldsymbol{W}\|_F^2 \right) \\
\leqslant & (2c \|\boldsymbol{W}\|_F^2)^2 = 4c^2 \|\boldsymbol{W}\|_F^4.
\end{aligned} \tag{C.143}$$

Similar to Lemma C.2.20, this coefficient is ℓ_1-norm bounded. The expression of each entry of $\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\mathsf{T}$ is just substituting $\mathbf{D}, \bar{\mathbf{D}}$ by $\mathbf{D}', \bar{\mathbf{D}}'$ in the polynomial characterized by Equation C.142. Therefore, using Lemma C.2.16, we have with probability 1 over $\boldsymbol{W}$, for all $i, j \in [c]$,

$$\lim_{n \to \infty} ((\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\mathsf{T})_{i,j} - (\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\mathsf{T})_{i,j}) = 0. \tag{C.144}$$

This completes the proof of the lemma as $\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\mathsf{T}$ is of constant size. □

Lemma C.2.22. *With probability 1 over* $\mathbf{W}^{(1)}$ *and* $\mathbf{W}^{(2)}$*,*

$$\lim_{n \to \infty} \frac{\left\| \boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\mathsf{T} \right\|_F^2}{\|\boldsymbol{M}^*\|_F^2} = 1. \tag{C.145}$$

Proof of Lemma C.2.22. The proof of this lemma will be divided into two parts. In the first part, we will estimate the Frobenius norm of $\boldsymbol{M}^*$, and in the second part

we do the same thing for $\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\mathsf{T}$.

Part 1: From Lemma C.2.19 we know that

$$\boldsymbol{M}^* = \frac{1}{4}\left(\boldsymbol{W}^\mathsf{T}\mathbb{E}[\mathbf{A}]\boldsymbol{W} + \mathit{diag}(\boldsymbol{W}^\mathsf{T}\mathbb{E}[\mathbf{A}]\boldsymbol{W})\right). \tag{C.146}$$

Denote $\widetilde{\mathbf{A}} \triangleq \mathbb{E}[\mathbf{A}]$, then

$$\mathbb{E}[\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}] = \boldsymbol{W}^\mathsf{T}\mathbb{E}[\mathbf{A}]\boldsymbol{W} = \boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}. \tag{C.147}$$

From Lemma C.2.5, for all $\epsilon' > 0$, with probability 1 over $\mathbf{W}$ we have $\left\|\boldsymbol{W}\boldsymbol{W}^\mathsf{T} - \boldsymbol{I}_c\right\| \leqslant \epsilon'$. Besides, from Kleinman and Athans (1968) we know that for positive semi-definite matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ we have $\lambda_{\min}(\boldsymbol{A})\operatorname{tr}(\boldsymbol{B}) \leqslant \operatorname{tr}(\boldsymbol{A}\boldsymbol{B}) \leqslant \lambda_{\max}(\boldsymbol{A})\operatorname{tr}(\boldsymbol{B})$, so

$$\begin{aligned}
\left|\left\|\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}\right\|_F^2 - \left\|\widetilde{\mathbf{A}}\right\|_F^2\right| &= \left|\operatorname{tr}(\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}) - \operatorname{tr}(\widetilde{\mathbf{A}}\widetilde{\mathbf{A}})\right| \\
&= \left|\operatorname{tr}(\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}) - \operatorname{tr}(\widetilde{\mathbf{A}}\widetilde{\mathbf{A}})\right| \\
&\leqslant \left|\left(\|\boldsymbol{W}\boldsymbol{W}^\mathsf{T} - \boldsymbol{I}_c\| + 1\right)\operatorname{tr}(\widetilde{\mathbf{A}}\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}) - \operatorname{tr}(\widetilde{\mathbf{A}}\widetilde{\mathbf{A}})\right| \\
&= \left|\left(\|\boldsymbol{W}\boldsymbol{W}^\mathsf{T} - \boldsymbol{I}_c\| + 1\right)\operatorname{tr}(\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\widetilde{\mathbf{A}}) - \operatorname{tr}(\widetilde{\mathbf{A}}\widetilde{\mathbf{A}})\right| \\
&\leqslant \left|\left(\|\boldsymbol{W}\boldsymbol{W}^\mathsf{T} - \boldsymbol{I}_c\| + 1\right)^2\operatorname{tr}(\widetilde{\mathbf{A}}\widetilde{\mathbf{A}}) - \operatorname{tr}(\widetilde{\mathbf{A}}\widetilde{\mathbf{A}})\right| \\
&\leqslant \|\boldsymbol{W}\boldsymbol{W}^\mathsf{T} - \boldsymbol{I}_c\|^2\left\|\widetilde{A}\right\|_F^2 + 2\|\boldsymbol{W}\boldsymbol{W}^\mathsf{T} - \boldsymbol{I}_c\|\left\|\widetilde{A}\right\|_F^2.
\end{aligned} \tag{C.148}$$

For any $\epsilon > 0$, set $\epsilon' = \min\{\frac{\epsilon}{4}, \frac{\sqrt{\epsilon}}{2}\}$ gives us with probability 1,

$$\lim_{n\to\infty} \frac{\left|\left\|\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}\right\|_F^2 - \left\|\widetilde{\mathbf{A}}\right\|_F^2\right|}{\left\|\widetilde{\mathbf{A}}\right\|_F^2} = 0, \tag{C.149}$$

i.e.,

$$\lim_{n\to\infty} \frac{\left\|\boldsymbol{W}^{\mathsf{T}}\widetilde{\mathbf{A}}\boldsymbol{W}\right\|_F^2}{\left\|\widetilde{\mathbf{A}}\right\|_F^2} = 1. \tag{C.150}$$

Besides, if we denote the i-th column of $\boldsymbol{W}$ by $\boldsymbol{w}_i$, then

$$\begin{aligned}\left\|\mathrm{diag}(\mathbb{E}[\boldsymbol{W}^{\mathsf{T}}\mathbf{A}\boldsymbol{W}])\right\|_F^2 &= \sum_{i=1}^{n}(\boldsymbol{w}_i^{\mathsf{T}}\widetilde{\mathbf{A}}\boldsymbol{w}_i)^2 \\ &\leqslant \sum_{i=1}^{n}\left(\|\boldsymbol{w}_i\|^2\cdot\|\widetilde{\mathbf{A}}\|\right)^2 \\ &= \left\|\widetilde{\mathbf{A}}\right\|^2\sum_{i=1}^{n}\|\boldsymbol{w}_i\|^4.\end{aligned} \tag{C.151}$$

Since $\mathbb{E}[n^2\|\boldsymbol{w}_i\|^4] = c^2+2c$, by the additive form of Chernoff bound we get

$$\Pr\left[\sum_{i=1}^{n}\|\boldsymbol{w}_i\|^4 \geqslant \frac{c^2+3c}{n}\right] = \Pr\left[\frac{\sum_{i=1}^{n}n^2\|\boldsymbol{w}_i\|^4}{n} - (c^2+2c) \geqslant c\right] \leqslant e^{-2nc^2}. \tag{C.152}$$

Therefore, when $n\to\infty$, with probability 1 over $\mathbf{W}$ we have

$$\left\|\mathrm{diag}(\mathbb{E}[\boldsymbol{W}^{\mathsf{T}}\mathbf{A}\boldsymbol{W}])\right\|_F^2 \leqslant \left\|\widetilde{\mathbf{A}}\right\|^2\sum_{i=1}^{n}\|\boldsymbol{w}_i\|^4 \leqslant \left\|\widetilde{\mathbf{A}}\right\|^2\cdot\frac{c^2+3c}{n}. \tag{C.153}$$

Thus, with probability 1 over $\mathbf{W}$,

$$\lim_{n\to\infty}\frac{\left\|\mathrm{diag}\left(\mathbb{E}\left[\boldsymbol{W}^{\mathsf{T}}\mathbf{A}\boldsymbol{W}\right]\right)\right\|_F^2}{\left\|\boldsymbol{W}^{\mathsf{T}}\widetilde{\mathbf{A}}\boldsymbol{W}\right\|_F^2} = 0, \tag{C.154}$$

i.e.,

$$\lim_{n\to\infty}\frac{\frac{1}{16}\left\|\widetilde{\mathbf{A}}\right\|_F^2}{\|\boldsymbol{M}^*\|_F^2} = 1. \tag{C.155}$$

Part 2: We now estimate the norm of $\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}$. Plug equation Equation C.133 into $\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}$ and we get

$$\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W} = \frac{1}{4}\left(\mathbb{E}[\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\boldsymbol{W}^\mathsf{T}] + \mathbb{E}[\boldsymbol{W}\mathrm{diag}(\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W})\boldsymbol{W}^\mathsf{T}]\right). \tag{C.156}$$

Similar to **Part 1**, when $n \to \infty$, with probability 1, we have

$$\lim_{n\to\infty} \frac{\left\|\mathbb{E}[\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\boldsymbol{W}^\mathsf{T}]\right\|_F^2}{\left\|\widetilde{\mathbf{A}}\right\|_F^2} = 1. \tag{C.157}$$

Besides, when $n \to \infty$, with probability 1 we have

$$\left\|\boldsymbol{W}\mathrm{diag}(\mathbb{E}[\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}])\boldsymbol{W}^\mathsf{T}\right\|_F^2 \leqslant \|\boldsymbol{W}\|_F^2\left\|\widetilde{\mathbf{A}}\right\|^2\sum_{i=1}^n\|\boldsymbol{w}_i\|^4 \leqslant \left\|\widetilde{\mathbf{A}}\right\|^2\cdot\frac{c^2+3c}{n}\|\boldsymbol{W}\|_F^2. \tag{C.158}$$

As a result, with probability 1,

$$\lim_{n\to\infty} \frac{\left\|\boldsymbol{W}\mathrm{diag}(\mathbb{E}[\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}])\boldsymbol{W}^\mathsf{T}\right\|_F^2}{\left\|\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\right\|_F^2} = 0, \tag{C.159}$$

i.e.,

$$\lim_{n\to\infty} \frac{\frac{1}{16}\left\|\widetilde{\mathbf{A}}\right\|_F^2}{\|\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\mathsf{T}\|_F^2} = 1. \tag{C.160}$$

Combining the results of **Part 1** and **Part 2** proves this lemma.

□

Combining Lemma C.2.20, Lemma C.2.21, and Lemma C.2.22 directly finishes the proof of Lemma C.2.18. □

After establishing the projection of $\boldsymbol{M}$ onto a $c \times c$ matrix, we may project the

full layer-wise Hessian of the first layer, namely $\boldsymbol{H}^{(1)} = \mathbb{E}[\mathbf{D}\boldsymbol{W}^{(2)\mathsf{T}}\mathbf{A}\boldsymbol{W}^{(2)}\mathbf{D} \otimes \mathbf{x}\mathbf{x}^{\mathsf{T}}]$ onto a $c \times c$ matrix using very similar techniques. For simplicity of notation, we will denote $\boldsymbol{W}^{(2)}$ by $\boldsymbol{W}$ and $\boldsymbol{H}^{(1)}$ by $\boldsymbol{H}$ unless explicitly stated otherwise.

Since the autocorrelation matrix $\mathbf{x}\mathbf{x}$ has unbounded Frobenious norm, we will consider a re-scaled version $\breve{\boldsymbol{H}} \triangleq \boldsymbol{H}/d^2$ for our analysis. Let $\boldsymbol{U} \triangleq \frac{1}{\sqrt{d}}\mathbf{1}_d^{\mathsf{T}} \in \mathbb{R}^{1\times d}$ be an all-1 matrix scaled by $\frac{1}{\sqrt{d}}$, we have $\boldsymbol{U}\boldsymbol{U}^{\mathsf{T}} = 1$. Let $\boldsymbol{V} \triangleq \boldsymbol{W} \otimes \boldsymbol{U}$ be our projection matrix for $\breve{\boldsymbol{H}}$, we may then state our main lemma for full layer-wise Hessian.

Lemma C.2.23. *With probability 1 over $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$,*

$$\lim_{n\to\infty} \frac{\left\|\boldsymbol{V}\breve{\boldsymbol{H}}\boldsymbol{V}^{\mathsf{T}}\right\|_F^2}{\left\|\breve{\boldsymbol{H}}\right\|_F^2} = 1. \tag{C.161}$$

Proof of Lemma C.2.23. Similar to the proof for the output Hessian, we will introduce a "bridging term"

$$\breve{\boldsymbol{H}}^* \triangleq \frac{1}{d}\mathbb{E}[\mathbf{D}'\boldsymbol{W}^{(2)\mathsf{T}}\mathbf{A}\boldsymbol{W}^{(2)}\mathbf{D}' \otimes \mathbf{x}''\mathbf{x}''^{\mathsf{T}}] \tag{C.162}$$

where $\mathbf{D}'$ is an independent copy of $\mathbf{D}$ and also independent of $\mathbf{A}$, and $\mathbf{x}''$ is an independent copy of $\mathbf{x}$ which is independent to both $\mathbf{D}'$ and $\mathbf{A}$. Informally, we will show

$$\left\|\boldsymbol{V}\breve{\boldsymbol{H}}\boldsymbol{V}^{\mathsf{T}}\right\|_F^2 \approx \left\|\boldsymbol{V}\breve{\boldsymbol{H}}^*\boldsymbol{V}^{\mathsf{T}}\right\|_F^2 \approx \left\|\breve{\boldsymbol{H}}^*\right\|_F^2 \approx \left\|\breve{\boldsymbol{H}}\right\|_F^2. \tag{C.163}$$

We first look into the structures of $\breve{\boldsymbol{H}}^*$.

Lemma C.2.24. *With probability 1 over $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$,*

$$\breve{\boldsymbol{H}}^* = \frac{1}{4d}\left(\boldsymbol{W}^{\mathsf{T}}\mathbb{E}[\mathbf{A}]\boldsymbol{W} + \operatorname{diag}(\boldsymbol{W}^{\mathsf{T}}\mathbb{E}[\mathbf{A}]\boldsymbol{W})\right) \otimes \left(\frac{1}{2\pi}\mathbf{1}_d\mathbf{1}_d^{\mathsf{T}} + \frac{\pi-1}{2\pi}\boldsymbol{I}_d\right). \tag{C.164}$$

Moreover, for large n, $\eta/32 < \left\|\breve{\boldsymbol{H}}^*\right\|_F < 2c^2$.

Proof of Lemma C.2.24. By independence in construction, we have $\breve{\boldsymbol{H}}^* = \boldsymbol{M}^* \otimes (\frac{1}{d}\mathbb{E}[\mathbf{x}''\mathbf{x}''^\mathsf{T}])$. Thus we only need to look into $\mathbb{E}[\mathbf{x}''\mathbf{x}''^\mathsf{T}]$. For $i = j$, we have $\mathbb{E}[\mathbf{x}''\mathbf{x}''^\mathsf{T}]_{ii} = \mathbb{E}[\mathbf{x}_i\mathbf{x}_i] = \frac{1}{2}$ while for $i \neq j$, $\mathbb{E}[\mathbf{x}''\mathbf{x}''^\mathsf{T}]_{ij} = \mathbb{E}[\mathbf{x}_i\mathbf{x}_j] = \frac{1}{2\pi}$. Thus

$$\mathbb{E}[\mathbf{x}''\mathbf{x}''^\mathsf{T}] = \frac{1}{2\pi}\mathbf{1}_d\mathbf{1}_d^\mathsf{T} + \frac{\pi-1}{2\pi}\boldsymbol{I}_d. \tag{C.165}$$

It follows that

$$\lim_{d\to\infty}\frac{1}{d}\left\|\mathbb{E}[\mathbf{x}\mathbf{x}^\mathsf{T}]\right\|_F = \lim_{d\to\infty}\frac{1}{d}\sqrt{d^2\frac{1}{4\pi^2} + d\frac{(\pi-1)^2}{4\pi^2}} = \frac{1}{2\pi} > \frac{1}{8}. \tag{C.166}$$

Thus for large n we have $\frac{1}{8} < \frac{1}{d}\left\|\mathbb{E}[\mathbf{x}\mathbf{x}^\mathsf{T}]\right\|_F < 1$. Since $\left\|\breve{\boldsymbol{H}}^*\right\|_F = \frac{1}{d}\left\|\boldsymbol{M}^* \otimes \mathbb{E}[\mathbf{x}\mathbf{x}^\mathsf{T}]\right\|_F = \left\|\boldsymbol{M}^*\right\|_F \cdot \frac{1}{d}\left\|\mathbb{E}[\mathbf{x}\mathbf{x}^\mathsf{T}]\right\|_F$ and we know that $\frac{\eta}{4} < \left\|\breve{\boldsymbol{H}}^*\right\|_F < 2c^2$ from Lemma C.2.19. We can conclude that for large n, $\eta/32 < \left\|\breve{\boldsymbol{H}}^*\right\|_F < 2c^2$. □

Lemma C.2.25. *With probability 1 over $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$,*

$$\lim_{n\to\infty}\frac{\left\|\breve{\boldsymbol{H}}\right\|_F^2}{\left\|\breve{\boldsymbol{H}}^*\right\|_F^2} = 1.$$

Proof of Lemma C.2.25. Unsurprisingly, this proof will be very similar to the proof of Lemma C.2.20. Recall that $\breve{\boldsymbol{H}}^* \triangleq \frac{1}{d}\mathbb{E}[\mathbf{D}'\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}' \otimes \mathbf{x}''\mathbf{x}''^\mathsf{T}]$. Let $(\bar{\mathbf{D}}, \bar{\mathbf{A}}, \bar{\mathbf{x}})$ be

an independent copy of $(\mathbf{D}, \mathbf{A}, \mathbf{x})$,

$$\begin{aligned}
\left\|\breve{\boldsymbol{H}}\right\|_F^2 &= \left\|\frac{1}{d}\mathbb{E}[\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}\otimes\mathbf{x}\mathbf{x}^\mathsf{T}]\right\|_F^2 \\
&= \mathbb{E}\left[\frac{1}{d^2}\langle \mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}\otimes\mathbf{x}\mathbf{x}^\mathsf{T}, \bar{\mathbf{D}}\boldsymbol{W}^\mathsf{T}\bar{\mathbf{A}}\boldsymbol{W}\bar{\mathbf{D}}\otimes\bar{\mathbf{x}}\bar{\mathbf{x}}^\mathsf{T}\rangle\right] \\
&= \mathbb{E}\left[\frac{1}{d^2}\operatorname{tr}\left(\left(\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}\otimes\mathbf{x}\mathbf{x}^\mathsf{T}\right)\left(\bar{\mathbf{D}}\boldsymbol{W}^\mathsf{T}\bar{\mathbf{A}}\boldsymbol{W}\bar{\mathbf{D}}\otimes\bar{\mathbf{x}}\bar{\mathbf{x}}^\mathsf{T}\right)\right)\right] \\
&= \mathbb{E}\left[\frac{1}{d^2}\operatorname{tr}\left(\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}\bar{\mathbf{D}}\boldsymbol{W}^\mathsf{T}\bar{\mathbf{A}}\boldsymbol{W}\bar{\mathbf{D}}\right)\operatorname{tr}\left(\mathbf{x}\mathbf{x}^\mathsf{T}\bar{\mathbf{x}}\bar{\mathbf{x}}^\mathsf{T}\right)\right] \\
&= \mathbb{E}\left[\frac{1}{d^2}(\mathbf{x}^\mathsf{T}\bar{\mathbf{x}}\bar{\mathbf{x}}^\mathsf{T}\mathbf{x})\operatorname{tr}\left(\boldsymbol{W}\bar{\mathbf{D}}\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}\bar{\mathbf{D}}\boldsymbol{W}^\mathsf{T}\bar{\mathbf{A}}\right)\right].
\end{aligned} \tag{C.167}$$

Expressing the term inside the expectation as a polynomial of entries of $\mathbf{A}$, $\mathbf{D}$, $\bar{\mathbf{A}}$ and $\bar{\mathbf{D}}$, we get

$$\begin{aligned}
&\frac{1}{d^2}(\mathbf{x}^\mathsf{T}\bar{\mathbf{x}}\bar{\mathbf{x}}^\mathsf{T}\mathbf{x})\operatorname{tr}\left(\boldsymbol{W}\bar{\mathbf{D}}\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}\bar{\mathbf{D}}\boldsymbol{W}^\mathsf{T}\bar{\mathbf{A}}\right) \\
=&\frac{1}{d^2}\sum_{p,q=1}^{d}\mathbf{x}_p\bar{\mathbf{x}}_p\mathbf{x}_q\bar{\mathbf{x}}_q\left(\sum_{i=1}^{c}\left(\boldsymbol{W}\bar{\mathbf{D}}\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}\bar{\mathbf{D}}\boldsymbol{W}^\mathsf{T}\bar{\mathbf{A}}\right)_{i,i}\right) \\
=&\frac{1}{d^2}\sum_{p,q=1}^{d}\sum_{i,j,k,s=1}^{c}\sum_{l,t=1}^{n}\boldsymbol{W}_{i,l}\boldsymbol{W}_{k,l}\boldsymbol{W}_{j,t}\boldsymbol{W}_{s,t}\bar{\mathbf{A}}_{k,j}\mathbf{A}_{s,i}\bar{\mathbf{D}}_{l,l}\mathbf{D}_{l,l}\bar{\mathbf{D}}_{t,t}\mathbf{D}_{t,t}\mathbf{x}_p\bar{\mathbf{x}}_p\mathbf{x}_q\bar{\mathbf{x}}_q.
\end{aligned} \tag{C.168}$$

We skipped some derivations as they are identical to Equation C.140. The monomials are $\bar{\mathbf{A}}_{k,j}\mathbf{A}_{s,i}\bar{\mathbf{D}}_{l,l}\mathbf{D}_{l,l}\bar{\mathbf{D}}_{t,t}\mathbf{D}_{t,t}\mathbf{x}_p\bar{\mathbf{x}}_p\mathbf{x}_q\bar{\mathbf{x}}_q$, and the corresponding coefficients are $\boldsymbol{W}_{i,l}\boldsymbol{W}_{k,l}\boldsymbol{W}_{j,t}\boldsymbol{W}_{s,t}$. The ℓ_1 norm of the coefficients is

$$\left\|\frac{1}{d^2}\sum_{p,q=1}^{d}\sum_{i,j,k,s=1}^{c}\sum_{l,t=1}^{n}\boldsymbol{W}_{i,l}\boldsymbol{W}_{k,l}\boldsymbol{W}_{j,t}\boldsymbol{W}_{s,t}\right\|_1 = \left\|\sum_{i,j,k,s=1}^{c}\sum_{l,t=1}^{n}\boldsymbol{W}_{i,l}\boldsymbol{W}_{k,l}\boldsymbol{W}_{j,t}\boldsymbol{W}_{s,t}\right\|_1, \tag{C.169}$$

which we know is upper bounded by some constant with probability 1 over $\boldsymbol{W}$ from

Equation C.141.

For any $\epsilon > 0$, fix ϵ. Note that $\left\|\breve{\boldsymbol{H}}^*\right\|_F^2$ is just substituting $(\mathbf{D}, \bar{\mathbf{D}}, \mathbf{x}, \bar{\mathbf{x}})$ by $(\mathbf{D}', \bar{\mathbf{D}}', \mathbf{x}'', \bar{\mathbf{x}}'')$ in the polynomial characterized by Equation C.168. From Corollary C.2.17 we have the convergence of the difference of the expectation of the two polynomials, namely $|\left\|\breve{\boldsymbol{H}}\right\|_F^2 - \left\|\breve{\boldsymbol{H}}^*\right\|_F^2| < \epsilon$ for sufficiently large n. Since the spectral norm of $\breve{\boldsymbol{H}}^*$ is bounded below from 0 by Lemma C.2.19, we have $\lim_{n\to\infty} \left\|\breve{\boldsymbol{H}}\right\|_F^2 / \left\|\breve{\boldsymbol{H}}^*\right\|_F^2 = 1$.

□

Lemma C.2.26. *For all $i, j \in [c]$, $\lim_{n\to\infty}((\boldsymbol{V}\breve{\boldsymbol{H}}\boldsymbol{V}^\mathsf{T})_{i,j} - (\boldsymbol{V}\breve{\boldsymbol{H}}^*\boldsymbol{V}^\mathsf{T})_{i,j}) = 0$. Thus,*

$$\lim_{n\to\infty} \frac{\left\|\boldsymbol{V}\breve{\boldsymbol{H}}\boldsymbol{V}^\mathsf{T}\right\|_F^2}{\left\|\boldsymbol{V}\breve{\boldsymbol{H}}^*\boldsymbol{V}^\mathsf{T}\right\|_F^2} = 1. \tag{C.170}$$

Proof of Lemma C.2.26. This proof is very similar to that of Lemma C.2.25. First, we focus on a single entry of the matrix $\boldsymbol{V}\breve{\boldsymbol{H}}\boldsymbol{V}^\mathsf{T}$ and express it as a polynomial of entries of $\mathbf{A}$ and $\mathbf{D}$:

$$\begin{aligned}
(\boldsymbol{V}\breve{\boldsymbol{H}}\boldsymbol{V}^\mathsf{T})_{i,j} &= \mathbb{E}\left[\left((\boldsymbol{W}\otimes\boldsymbol{U})\frac{1}{d}(\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}\otimes\mathbf{x}\mathbf{x}^\mathsf{T})(\boldsymbol{W}\otimes\boldsymbol{U})^\mathsf{T}\right)_{i,j}\right] \\
&= \mathbb{E}\left[\frac{1}{d}\left((\boldsymbol{W}\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}\boldsymbol{W}^\mathsf{T})\otimes(\boldsymbol{U}\mathbf{x}\mathbf{x}^\mathsf{T}\boldsymbol{U}^\mathsf{T})\right)_{i,j}\right] \\
&= \mathbb{E}\left[\frac{1}{d}\cdot\frac{1}{d}(\mathbf{1}_d^\mathsf{T}\mathbf{x}\mathbf{x}^\mathsf{T}\mathbf{1}_d)\left(\boldsymbol{W}\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}\boldsymbol{W}^\mathsf{T}\right)_{i,j}\right] \\
&= \mathbb{E}\left[\frac{1}{d^2}\left(\sum_{p,q=1}^{d}\mathbf{x}_p\mathbf{x}_q\right)\left(\sum_{k=1}^{c}(\boldsymbol{W}\mathbf{D}\boldsymbol{W}^\mathsf{T}\mathbf{A})_{i,k}(\boldsymbol{W}\mathbf{D}\boldsymbol{W}^\mathsf{T})_{k,j}\right)\right] \\
&= \mathbb{E}\left[\frac{1}{d^2}\sum_{p,q=1}^{d}\sum_{k,s=1}^{c}\sum_{l,t=1}^{n}\boldsymbol{W}_{i,l}\boldsymbol{W}_{s,l}\boldsymbol{W}_{k,t}\boldsymbol{W}_{j,t}\mathbf{A}_{s,k}\mathbf{D}_{l,l}\mathbf{D}_{t,t}\mathbf{x}_p\mathbf{x}_q\right].
\end{aligned} \tag{C.171}$$

We skipped some derivations as they are identical to Equation C.142. The monomials are $\mathbf{A}_{s,k}\mathbf{D}_{l,l}\mathbf{D}_{t,t}\mathbf{x}_p\mathbf{x}_q$, and the corresponding coefficients are $\boldsymbol{W}_{i,l}\boldsymbol{W}_{s,l}\boldsymbol{W}_{k,t}\boldsymbol{W}_{j,t}$. Observe that the ℓ_1 norm of the coefficients satisfies

$$\left\| \frac{1}{d^2} \sum_{p,q=1}^{d} \sum_{k,s=1}^{c} \sum_{l,t=1}^{n} \boldsymbol{W}_{i,l}\boldsymbol{W}_{s,l}\boldsymbol{W}_{k,t}\boldsymbol{W}_{j,t} \right\|_1 = \left\| \sum_{k,s=1}^{c} \sum_{l,t=1}^{n} \boldsymbol{W}_{i,l}\boldsymbol{W}_{s,l}\boldsymbol{W}_{k,t}\boldsymbol{W}_{j,t} \right\|_1, \quad \text{(C.172)}$$

which we know is bounded above by some constant from Equation C.143. Note that the expression of each entry of $\boldsymbol{V}\breve{\boldsymbol{H}}^*\boldsymbol{W}^\mathsf{T}$ is just substituting $(\mathbf{D}, \bar{\mathbf{D}}, \mathbf{x}, \bar{\mathbf{x}})$ by $(\mathbf{D}', \bar{\mathbf{D}}', \mathbf{x}'', \bar{\mathbf{x}}'')$ in the polynomial characterized by Equation C.171. Therefore, using Lemma C.2.16, we have with probability 1 over $\boldsymbol{W}$, for all $i, j \in [c]$,

$$\lim_{n\to\infty} ((\boldsymbol{V}\breve{\boldsymbol{H}}\boldsymbol{V}^\mathsf{T})_{i,j} - (\boldsymbol{V}\breve{\boldsymbol{H}}^*\boldsymbol{V}^\mathsf{T})_{i,j}) = 0. \quad \text{(C.173)}$$

This completes the proof of the lemma as $\boldsymbol{V}\breve{\boldsymbol{H}}\boldsymbol{V}^\mathsf{T}$ is of constant size. □

Lemma C.2.27. *With probability 1 over* $\mathbf{W}^{(1)}$ *and* $\mathbf{W}^{(2)}$*,*

$$\lim_{n\to\infty} \frac{\left\| \boldsymbol{V}\breve{\boldsymbol{H}}^*\boldsymbol{V}^\mathsf{T} \right\|_F^2}{\left\| \breve{\boldsymbol{H}}^* \right\|_F^2} = 1. \quad \text{(C.174)}$$

Proof of Lemma C.2.27. This lemma is a direct corollary of Lemma C.2.22 for

the output Hessian. Note that by the independence in construction,

$$\begin{aligned}
\boldsymbol{V}\breve{\boldsymbol{H}}^*\boldsymbol{V}^\mathsf{T} &= \frac{1}{d}\left(\boldsymbol{W}\otimes\boldsymbol{U}\right)\mathbb{E}\left[\mathbf{D}'\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}\mathbf{D}\otimes\mathbf{x}''\mathbf{x}''^\mathsf{T}\right]\left(\boldsymbol{W}^\mathsf{T}\otimes\boldsymbol{U}^\mathsf{T}\right) \\
&= \frac{1}{d}\left(\boldsymbol{W}\otimes\boldsymbol{U}\right)\left(\boldsymbol{M}^*\otimes\mathbb{E}\left[\mathbf{x}''\mathbf{x}''^\mathsf{T}\right]\right)\left(\boldsymbol{W}^\mathsf{T}\otimes\boldsymbol{U}^\mathsf{T}\right) \\
&= \frac{1}{d}\left(\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\mathsf{T}\right)\otimes\left(\boldsymbol{U}\mathbb{E}\left[\mathbf{x}''\mathbf{x}''^\mathsf{T}\right]\boldsymbol{U}^\mathsf{T}\right) \\
&= \left(\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\mathsf{T}\right)\otimes\left(\frac{1}{d^2}\mathbf{1}_d^\mathsf{T}\mathbb{E}\left[\mathbf{x}''\mathbf{x}''^\mathsf{T}\right]\mathbf{1}_d\right) \\
&= \frac{1}{d^2}\mathbf{1}_d^\mathsf{T}\mathbb{E}\left[\mathbf{x}''\mathbf{x}''^\mathsf{T}\right]\mathbf{1}_d\left(\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\mathsf{T}\right).
\end{aligned} \tag{C.175}$$

From Equation C.165 we have

$$\mathbf{1}_d^\mathsf{T}\mathbb{E}\left[\mathbf{x}''\mathbf{x}''^\mathsf{T}\right]\mathbf{1}_d = \sum_{i,j=1}^{d}\mathbb{E}\left[\mathbf{x}\mathbf{x}^\mathsf{T}\right]_{ij} = \frac{1}{2\pi}d^2 + \frac{\pi-1}{2\pi}d. \tag{C.176}$$

Thus

$$\begin{aligned}
\left\|\boldsymbol{V}\breve{\boldsymbol{H}}^*\boldsymbol{V}^\mathsf{T}\right\|_F^2 &= \left\|\left(\frac{1}{2\pi}+\frac{\pi-1}{2\pi d}\right)\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}\right\|_F^2 \\
&= \left(\frac{1}{4\pi^2}+\frac{\pi-1}{2\pi^2 d}+\frac{(\pi-1)^2}{4\pi^2 d^2}\right)\|\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}\|_F^2.
\end{aligned} \tag{C.177}$$

Meanwhile note that

$$\left\|\breve{\boldsymbol{H}}^*\right\|_F^2 = \frac{1}{d^2}\left\|\widetilde{\boldsymbol{M}}^*\otimes\mathbb{E}\left[\mathbf{x}''\mathbf{x}''^\mathsf{T}\right]\right\|_F^2 = \frac{1}{d^2}\left\|\widetilde{\boldsymbol{M}}^*\right\|_F^2\otimes\left\|\mathbb{E}\left[\mathbf{x}''\mathbf{x}''^\mathsf{T}\right]\right\|_F^2, \tag{C.178}$$

where

$$\left\|\mathbb{E}\left[\mathbf{x}''\mathbf{x}''^\mathsf{T}\right]\right\|_F^2 = \sum_{i,j=1}^{d}\mathbb{E}\left[\mathbf{x}\mathbf{x}^\mathsf{T}\right]_{ij}^2 = \frac{1}{4\pi^2}d^2 + \frac{\pi-1}{2\pi}d. \tag{C.179}$$

Thus

$$\left\|\breve{\boldsymbol{H}}^*\right\|_F^2 = \left(\frac{1}{4\pi^2} + \frac{\pi-1}{2\pi d}\right)\left\|\widetilde{\boldsymbol{M}}^*\right\|_F^2. \tag{C.180}$$

Since $d = n^{1+\alpha}$ for some constant $\alpha > 0$, we have

$$\lim_{n\to\infty} \frac{\frac{1}{4\pi^2} + \frac{\pi-1}{2\pi^2 d} + \frac{(\pi-1)^2}{4\pi^2 d^2}}{\frac{1}{4\pi^2} + \frac{\pi-1}{2\pi d}} = 1. \tag{C.181}$$

Thus combined with the result from Lemma C.2.22, we have

$$\lim_{n\to\infty} \frac{\left\|\boldsymbol{V}\breve{\boldsymbol{H}}^*\boldsymbol{V}^\mathsf{T}\right\|_F^2}{\left\|\breve{\boldsymbol{H}}^*\right\|_F^2} = \left(\lim_{n\to\infty} \frac{\frac{1}{4\pi^2} + \frac{\pi-1}{2\pi^2 d} + \frac{(\pi-1)^2}{4\pi^2 d^2}}{\frac{1}{4\pi^2} + \frac{\pi-1}{2\pi d}}\right)\left(\lim_{n\to\infty} \frac{\left\|\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\mathsf{T}\right\|_F^2}{\left\|\boldsymbol{M}^*\right\|_F^2}\right) = 1. \tag{C.182}$$

□

Combining Lemma C.2.25, Lemma C.2.26, and Lemma C.2.27 completes the proof of Lemma C.2.23. □

Now we are done with the lemmas and will proceed to the proof of the main theorems.

Structure of Output Hessian of the First Layer

We first restate Theorem 4.4.2 here:

Theorem 4.4.2 *Let* $\boldsymbol{M}^* \triangleq \mathbb{E}\left[\mathbf{D}'\boldsymbol{W}^{(2)\mathsf{T}}\mathbf{A}\boldsymbol{W}^{(2)}\mathbf{D}'\right]$ *where* $\mathbf{D}'$ *is an independent copy of* $\mathbf{D}$ *and is also independent of* $\mathbf{A}$*. Let* S_1 *and* S_2 *be the top* $c-1$ *eigenspaces of* $\boldsymbol{M}^{(1)}$ *and* $\boldsymbol{M}^*$ *respectively, for all* $\epsilon > 0$*,*

$$\lim_{n\to\infty} \Pr_{\boldsymbol{W}^{(1)}\sim\mathcal{N}(0,\frac{1}{d}\boldsymbol{I}_{nd}),\boldsymbol{W}^{(2)}\sim\mathcal{N}(0,\frac{1}{n}\boldsymbol{I}_{cn})}\left[\text{Overlap}\left(S_1, S_2\right) > 1-\epsilon\right] = 1. \tag{C.183}$$

Moreover, as

$$\lim_{n\to\infty} \Pr_{\boldsymbol{W}^{(1)}\sim\mathcal{N}(0,\frac{1}{d}\boldsymbol{I}_{nd}),\boldsymbol{W}^{(2)}\sim\mathcal{N}(0,\frac{1}{n}\boldsymbol{I}_{cn})}\left[\left(\frac{\lambda_c(\boldsymbol{M})}{\lambda_{c-1}(\boldsymbol{M})}\Big|_{\boldsymbol{W}^{(1)},\boldsymbol{W}^{(2)}}\right) < \epsilon\right] = 1. \tag{C.184}$$

Proof of Theorem 4.4.2. □

From Lemma C.2.18 we have

$$\lim_{n\to\infty} \frac{\left\|\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\right\|_F^2}{\|\boldsymbol{M}\|_F^2} = 1. \tag{C.185}$$

Then we consider $\left\|\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\right\|_F^2$. Note that

$$\begin{aligned}\left\|\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\right\|_F^2 &= \operatorname{tr}(\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W})\\ &= \operatorname{tr}(\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top).\end{aligned} \tag{C.186}$$

From Lemma C.2.5 we know that for all $\epsilon' > 0$, $\lim_{n\to\infty}\Pr\left(\left\|\boldsymbol{W}\boldsymbol{W}^\top - \boldsymbol{I}_c\right\| \geqslant \epsilon'\right) = 0$. For notation simplicity, in this proof we will omit the limit and probability arguments which can be dealt with using union bound. Therefore, we will directly state $\left\|\boldsymbol{W}\boldsymbol{W}^\top - \boldsymbol{I}_c\right\| \leqslant \epsilon'$. From Kleinman and Athans (1968) we know that for positive semi-definite matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ we have $\lambda_{\min}(\boldsymbol{A})\operatorname{tr}(\boldsymbol{B}) \leqslant \operatorname{tr}(\boldsymbol{A}\boldsymbol{B}) \leqslant \lambda_{\max}(\boldsymbol{A})\operatorname{tr}(\boldsymbol{B})$, so

$$\begin{aligned}&\left|\operatorname{tr}(\boldsymbol{W}\boldsymbol{W}^\top\cdot\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top) - \operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top)\right|\\ \leqslant&\max\{1-\lambda_{\min}(\boldsymbol{W}\boldsymbol{W}^\top), \lambda_{\max}(\boldsymbol{W}\boldsymbol{W}^\top)-1\}\operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top)\\ \leqslant&\left\|\boldsymbol{W}\boldsymbol{W}^\top - \boldsymbol{I}_c\right\|\operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top) \leqslant \epsilon'\operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top).\end{aligned} \tag{C.187}$$

Similarly,

$$\begin{aligned}
&|\operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top) - \operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top)| \\
=&|\operatorname{tr}(\boldsymbol{W}\boldsymbol{W}^\top\cdot\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top) - \operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top)| \\
\leqslant&\big\|\boldsymbol{W}\boldsymbol{W}^\top - \boldsymbol{I}_c\big\|\operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top) \leqslant \epsilon'\operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top).
\end{aligned} \tag{C.188}$$

Therefore,

$$\begin{aligned}
&|\big\|\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\big\|_F^2 - \big\|\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\big\|_F^2| \\
=&|\operatorname{tr}(\boldsymbol{W}\boldsymbol{W}^\top\cdot\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top) - \operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top)| \\
\leqslant&|\operatorname{tr}(\boldsymbol{W}\boldsymbol{W}^\top\cdot\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top) - \operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top)| \\
&+|\operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top) - \operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top)| \\
\leqslant&\epsilon'\operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top) + \epsilon'\operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top) \\
\leqslant&\epsilon'(1+\epsilon')\operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top) + \epsilon'\operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top) \\
\leqslant&(2\epsilon' + (\epsilon')^2)\operatorname{tr}(\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top) = (2\epsilon' + (\epsilon')^2)\big\|\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\big\|_F^2.
\end{aligned} \tag{C.189}$$

For all $\epsilon > 0$, select $\epsilon' < \min\{\frac{\sqrt{\epsilon}}{2}, \frac{\epsilon}{4}\}$, we have

$$|\big\|\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\big\|_F^2 - \big\|\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\big\|_F^2| < \epsilon\big\|\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\big\|_F^2. \tag{C.190}$$

In other words,

$$\lim_{n\to\infty}\frac{\big\|\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\big\|_F^2}{\|\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\|_F^2} = 1. \tag{C.191}$$

Hence we get

$$\lim_{n\to\infty}\frac{\big\|\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\big\|_F^2}{\|\boldsymbol{M}\|_F^2} = 1. \tag{C.192}$$

Next, consider the orthogonal projection matrix $P_{\boldsymbol{W}} \triangleq \overline{\boldsymbol{W}}^\top\overline{\boldsymbol{W}}$ that projects vectors

in $\mathbb{R}^n$ into the subspace spanned by all rows of $\boldsymbol{W}$. Here $\overline{\boldsymbol{W}}$ is the orthogonolized $\boldsymbol{W}$, which is explicitly defined in Lemma C.2.8. We will consider the matrix $P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}$. Define $\delta \triangleq \boldsymbol{W}^\top\boldsymbol{W} - P_{\boldsymbol{W}}$, then from Lemma C.2.8 we get $\|\delta\|_F^2 \leqslant \epsilon'$. Therefore,

$$\begin{aligned}
&\left|\left\|\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\right\|_F - \|P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}\|_F\right| \\
\leqslant\ &\|P_{\boldsymbol{W}}\boldsymbol{M}\delta\|_F + \|\delta\boldsymbol{M}P_{\boldsymbol{W}}\|_F + \|\delta\boldsymbol{M}\delta\|_F \\
\leqslant\ &\|\boldsymbol{M}\|_F \Pr\left[2\|P_{\boldsymbol{W}}\|_F\|\delta\|_F + \|\delta\|_F^2\right] \\
\leqslant\ &\|\boldsymbol{M}\|_F \Pr\left[2\cdot 4c^2\epsilon' + (\epsilon')^2\right].
\end{aligned} \tag{C.193}$$

For all $\epsilon > 0$, we choose $\epsilon' < \min\{\frac{\sqrt{\epsilon}}{2}, \frac{\epsilon}{16c^2}\}$ and have

$$\frac{\left|\left\|\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\right\|_F - \|P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}\|_F\right|}{\|\boldsymbol{M}\|_F} < \epsilon, \tag{C.194}$$

which means that

$$\begin{aligned}
&\lim_{n\to\infty} \frac{\left|\left\|\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\right\|_F - \|P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}\|_F\right|}{\left\|\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\right\|_F} \\
=\ &\lim_{n\to\infty} \frac{\left|\left\|\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\right\|_F - \|P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}\|_F\right|}{\|\boldsymbol{M}\|_F} = 0.
\end{aligned} \tag{C.195}$$

Thus,

$$\lim_{n\to\infty} \frac{\|P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}\|_F}{\|\boldsymbol{M}\|_F} = \lim_{n\to\infty} \frac{\|P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}\|_F}{\|\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^\top\boldsymbol{W}\|_F} = 1. \tag{C.196}$$

Note that $\|\boldsymbol{M}\|_F^2 = \|P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}\|_F^2 + \left\|P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}^\perp\right\|_F^2 + \left\|P_{\boldsymbol{W}}^\perp\boldsymbol{M}P_{\boldsymbol{W}}\right\|_F^2 + \left\|P_{\boldsymbol{W}}^\perp\boldsymbol{M}P_{\boldsymbol{W}}^\perp\right\|_F^2$. It follows that,

$$\begin{aligned}
&\lim_{n\to\infty} \frac{\left\|P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}^\perp\right\|_F^2 + \left\|P_{\boldsymbol{W}}^\perp\boldsymbol{M}P_{\boldsymbol{W}}\right\|_F^2 + \left\|P_{\boldsymbol{W}}^\perp\boldsymbol{M}P_{\boldsymbol{W}}^\perp\right\|_F^2}{\|\boldsymbol{M}\|_F^2} \\
=\ &\lim_{n\to\infty} \frac{\|\boldsymbol{M}\|_F^2 - \|P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}\|_F^2}{\|\boldsymbol{M}\|_F^2} = 0.
\end{aligned} \tag{C.197}$$

In other words,

$$\lim_{n\to\infty} \frac{\left\|P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}^{\perp}\right\|_F}{\|\boldsymbol{M}\|_F} = \lim_{n\to\infty} \frac{\left\|P_{\boldsymbol{W}}^{\perp}\boldsymbol{M}P_{\boldsymbol{W}}\right\|_F}{\|\boldsymbol{M}\|_F} = \lim_{n\to\infty} \frac{\left\|P_{\boldsymbol{W}}^{\perp}\boldsymbol{M}P_{\boldsymbol{W}}^{\perp}\right\|_F}{\|\boldsymbol{M}\|_F} = 0. \tag{C.198}$$

From Lemma C.2.19 we know that for large n, $\lim_{n\to\infty}\|\boldsymbol{M}\|_F$ is lower bounded by some constant that is independent of n, so

$$\lim_{n\to\infty} \left\|P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}^{\perp}\right\|_F = \lim_{n\to\infty} \left\|P_{\boldsymbol{W}}^{\perp}\boldsymbol{M}P_{\boldsymbol{W}}\right\|_F = \lim_{n\to\infty} \left\|P_{\boldsymbol{W}}^{\perp}\boldsymbol{M}P_{\boldsymbol{W}}^{\perp}\right\|_F = 0. \tag{C.199}$$

Note that

$$\boldsymbol{M} = P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}} + P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}^{\perp} + P_{\boldsymbol{W}}^{\perp}\boldsymbol{M}P_{\boldsymbol{W}} + P_{\boldsymbol{W}}^{\perp}\boldsymbol{M}P_{\boldsymbol{W}}^{\perp}. \tag{C.200}$$

Thus,

$$\lim_{n\to\infty} \|\boldsymbol{M} - P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}}\|_F = 0. \tag{C.201}$$

For any $\epsilon > 0$, set $\delta < \min\{\frac{\epsilon\eta}{8c^2}, \frac{\sqrt{\epsilon\eta}}{2c}\}$, from Lemma C.2.8, we know that with probability 1, $\left\|P_{\boldsymbol{W}} - \boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_F \leqslant \delta$. Therefore,

$$\begin{aligned}
&\left\|P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}} - \boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_F \\
\leqslant &\left\|P_{\boldsymbol{W}} - \boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_F^2\|\boldsymbol{M}\|_F + 2\left\|P_{\boldsymbol{W}} - \boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_F\|\boldsymbol{M}\|_F\|P_{\boldsymbol{W}}\|_F \\
\leqslant &\delta^2 \cdot 2c^2 + 2\delta \cdot 2c^2 \\
< &\epsilon.
\end{aligned} \tag{C.202}$$

In other words,

$$\lim_{n\to\infty} \left\|P_{\boldsymbol{W}}\boldsymbol{M}P_{\boldsymbol{W}} - \boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_F = 0. \tag{C.203}$$

Now we conclude that

$$\lim_{n\to\infty} \left\|\boldsymbol{M} - \boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_F = 0. \tag{C.204}$$

From Lemma C.2.21 we know that

$$\lim_{n\to\infty} \left\|\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^{\mathsf{T}} - \boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\right\|_{F} = 0. \tag{C.205}$$

Since

$$\left\|\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W} - \boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_{F} \leqslant \|\boldsymbol{W}\|_{F}^{2}\left\|\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^{\mathsf{T}} - \boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\right\|_{F}, \tag{C.206}$$

from Lemma C.2.3 which bounds the Frobenius norm of $\boldsymbol{W}$ we know that

$$\lim_{n\to\infty} \left\|\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W} - \boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_{F} = 0. \tag{C.207}$$

Thus,

$$\lim_{n\to\infty} \left\|\boldsymbol{M} - \boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_{F} = 0. \tag{C.208}$$

Note that $\boldsymbol{M}^{*} = \frac{1}{4}\left(\mathbb{E}[\boldsymbol{W}^{\mathsf{T}}\mathbf{A}\boldsymbol{W}] + \operatorname{diag}(\mathbb{E}[\boldsymbol{W}^{\mathsf{T}}\mathbf{A}\boldsymbol{W}])\right)$, so

$$4\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W} = \boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{W}^{\mathsf{T}}\widetilde{\mathbf{A}}\boldsymbol{W}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W} + \boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\operatorname{diag}(\mathbb{E}[\boldsymbol{W}^{\mathsf{T}}\mathbf{A}\boldsymbol{W}])\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}. \tag{C.209}$$

We will first analyze the second term on the RHS of equation Equation C.209. For all $\epsilon > 0$, set $\epsilon' = \frac{\epsilon}{\sqrt{c}}$, and from Lemma C.2.5 we know that $\left\|\boldsymbol{W}\boldsymbol{W}^{\mathsf{T}} - \boldsymbol{I}_{c}\right\| < \epsilon'$ with probability 1, which means that $\left|\left\|\boldsymbol{W}\boldsymbol{W}^{\mathsf{T}}\right\|_{F} - c\right| < \epsilon$ with probability 1. Set $\epsilon = c$, we know that $\left\|\boldsymbol{W}\boldsymbol{W}^{\mathsf{T}}\right\|_{F} < 2c$ with probability 1. Note that

$$\begin{aligned}
\left\|\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\operatorname{diag}(\mathbb{E}[\boldsymbol{W}^{\mathsf{T}}\mathbf{A}\boldsymbol{W}])\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_{F} &\leqslant \left\|\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_{F}^{2}\left\|\operatorname{diag}(\mathbb{E}[\boldsymbol{W}^{\mathsf{T}}\mathbf{A}\boldsymbol{W}])\right\|_{F} \\
&= \left\|\boldsymbol{W}\boldsymbol{W}^{\mathsf{T}}\right\|_{F}^{2}\left\|\operatorname{diag}(\mathbb{E}[\boldsymbol{W}^{\mathsf{T}}\mathbf{A}\boldsymbol{W}])\right\|_{F} \\
&\leqslant 4c^{2}\left\|\operatorname{diag}(\mathbb{E}[\boldsymbol{W}^{\mathsf{T}}\mathbf{A}\boldsymbol{W}])\right\|_{F}.
\end{aligned} \tag{C.210}$$

Combine this with equation Equation C.154 and we have

$$\lim_{n\to\infty} \frac{\left\|\boldsymbol{W}^\mathsf{T}\boldsymbol{W}\mathrm{diag}(\mathbb{E}[\boldsymbol{W}^\mathsf{T}\mathbf{A}\boldsymbol{W}])\boldsymbol{W}^\mathsf{T}\boldsymbol{W}\right\|_F}{\left\|\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}\right\|_F} = 0. \tag{C.211}$$

From Lemma C.2.19 we know that $\left\|\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}\right\|_F \geqslant \frac{\eta}{4}$ with probability 1, so

$$\lim_{n\to\infty} \left\|4\boldsymbol{W}^\mathsf{T}\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\mathsf{T}\boldsymbol{W} - \boldsymbol{W}^\mathsf{T}\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\boldsymbol{W}\right\|_F = 0. \tag{C.212}$$

Similarly, define $\delta \triangleq \boldsymbol{W}\boldsymbol{W}^\mathsf{T} - \boldsymbol{I}_c$, then

$$\begin{aligned}
&\left\|\boldsymbol{W}^\mathsf{T}\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\boldsymbol{W} - \boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}\right\|_F \\
\leqslant &\left\|\boldsymbol{W}^\mathsf{T}\delta\widetilde{\mathbf{A}}\delta\boldsymbol{W}\right\|_F + 2\left\|\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\delta\right\|_F \\
\leqslant &\|\boldsymbol{W}\|_F^2\|\delta\|_F^2\left\|\widetilde{\mathbf{A}}\right\|_F + 2\|\boldsymbol{W}\|_F\|\delta\|_F\left\|\widetilde{\mathbf{A}}\right\|_F.
\end{aligned} \tag{C.213}$$

Set $\epsilon' < \min\{\frac{\epsilon}{8c^2}, \sqrt{\frac{\epsilon}{8c^3}}\}$, then from Lemma C.2.5 we know that $\|\delta\|_F < \epsilon'$ with probability 1, and from Lemma C.2.3 we have $\|\boldsymbol{W}\|_F \leqslant 2c$ with probability 1. We also have $\left\|\widetilde{\mathbf{A}}\right\|_F \leqslant c$ since each entry of $\mathbf{A}$ is bounded by 1 in absolute value. Therefore,

$$\left\|\boldsymbol{W}^\mathsf{T}\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\boldsymbol{W} - \boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}\right\|_F \leqslant 4c^2(\epsilon')^2\cdot c + 2\cdot 2c\epsilon'\cdot c < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon, \tag{C.214}$$

which means that

$$\lim_{n\to\infty} \left\|\boldsymbol{W}^\mathsf{T}\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}\boldsymbol{W}^\mathsf{T}\boldsymbol{W} - \boldsymbol{W}^\mathsf{T}\widetilde{\mathbf{A}}\boldsymbol{W}\right\|_F = 0. \tag{C.215}$$

From Equation C.212 and Equation C.215 we get

$$\lim_{n\to\infty}\left\|\frac{1}{4}\boldsymbol{W}^{\mathsf{T}}\widetilde{\mathbf{A}}\boldsymbol{W}-\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_{F}=0. \tag{C.216}$$

Combining with Equation C.208 we have

$$\lim_{n\to\infty}\left\|\boldsymbol{M}-\frac{1}{4}\boldsymbol{W}^{\mathsf{T}}\widetilde{\mathbf{A}}\boldsymbol{W}\right\|_{F}=0. \tag{C.217}$$

Besides, from equation Equation C.64 in Lemma C.2.8 we know that for any $\epsilon'>0$,

$$\left\|\overline{\boldsymbol{W}}-\boldsymbol{W}\right\|_{F}^{2}=\sum_{i\in[c]}\left\|\overline{\boldsymbol{W}}_{i}-\boldsymbol{W}_{i}\right\|^{2}<\epsilon', \tag{C.218}$$

where $\overline{\boldsymbol{W}}$ is the orthogonal version of $\boldsymbol{W}$, i.e., we run the Gram-Schmidt process for the rows of $\boldsymbol{W}$. Define $\delta\triangleq\overline{\boldsymbol{W}}-\boldsymbol{W}$, for any $\epsilon>0$, set $\epsilon'=\min\{\frac{\epsilon}{8c^{2}},\sqrt{\frac{\epsilon}{2c}}\}$, we have with probability 1,

$$\begin{aligned}\left\|\boldsymbol{W}^{\mathsf{T}}\widetilde{\mathbf{A}}\boldsymbol{W}-\overline{\boldsymbol{W}}^{\mathsf{T}}\widetilde{\mathbf{A}}\overline{\boldsymbol{W}}\right\|_{F}&\leqslant 2\|\delta\|_{F}\left\|\widetilde{\mathbf{A}}\right\|_{F}\|\boldsymbol{W}\|_{F}+\|\delta\|_{F}^{2}\left\|\widetilde{\mathbf{A}}\right\|_{F}\\&\leqslant 4c^{2}\epsilon'+c(\epsilon')^{2}<\epsilon.\end{aligned} \tag{C.219}$$

Therefore,

$$\lim_{n\to\infty}\left\|\boldsymbol{W}^{\mathsf{T}}\widetilde{\mathbf{A}}\boldsymbol{W}-\overline{\boldsymbol{W}}^{\mathsf{T}}\widetilde{\mathbf{A}}\overline{\boldsymbol{W}}\right\|_{F}=0, \tag{C.220}$$

which implies

$$\lim_{n\to\infty}\left\|\boldsymbol{M}-\frac{1}{4}\overline{\boldsymbol{W}}^{\mathsf{T}}\widetilde{\mathbf{A}}\overline{\boldsymbol{W}}\right\|_{F}=0. \tag{C.221}$$

From Lemma C.2.15 we know that with probability 1, $\widetilde{\mathbf{A}}$ is of rank $(c-1)$. Since $\mathbf{A}\cdot\mathbf{1}=0$ is always true, the top $(c-1)$ eigenspace of $\widetilde{\mathbf{A}}$ is $\mathbb{R}^{c}\backslash\{\mathbf{1}\}$. Note that the rows in $\overline{\boldsymbol{W}}$ are of unit norm and orthogonal to each other, we conclude that $\overline{\boldsymbol{W}}^{\mathsf{T}}\widetilde{\mathbf{A}}\overline{\boldsymbol{W}}$ is

of rank $(c-1)$ and the corresponding eigenspace is $\mathcal{R}\{\overline{\boldsymbol{W}}_i\}_{i=1}^c \backslash \{\mathbf{1}^\mathsf{T}\overline{\boldsymbol{W}}\}$. Moreover, the minimum positive eigenvalue of $\overline{\boldsymbol{W}}^\mathsf{T}\widetilde{\mathbf{A}}\overline{\boldsymbol{W}}$ is lower bounded by $\frac{\eta}{4}$.

As for the top $c-1$ eigenvectors of $\boldsymbol{M}$, define $\delta \triangleq \boldsymbol{M} - \frac{1}{4}\overline{\boldsymbol{W}}^\mathsf{T}\widetilde{\mathbf{A}}\overline{\boldsymbol{W}}$, then $\boldsymbol{M} = \frac{1}{4}\overline{\boldsymbol{W}}^\mathsf{T}\widetilde{\mathbf{A}}\overline{\boldsymbol{W}} + \delta$. Define S_1 as the top $c-1$ eigenspaces for $\boldsymbol{M}$, and S_2 to be the top $c-1$ eigenspaces for $\frac{1}{4}\overline{\boldsymbol{W}}^\mathsf{T}\widetilde{\mathbf{A}}\overline{\boldsymbol{W}}$. Then from Davis-Kahan Theorem we know that

$$\left\|\sin\Theta(S_1, S_2)\right\|_F \leqslant \frac{\|\delta\|_F}{\lambda_{c-1}(\frac{1}{4}\overline{\boldsymbol{W}}^\mathsf{T}\widetilde{\mathbf{A}}\overline{\boldsymbol{W}})}. \tag{C.222}$$

Here $\Theta(S_1, S_2)$ is a $(c-1)\times(c-1)$ diagonal matrix whose i-th diagonal entry is the i-th canonical angle between S_1 and S_2. Since $\lim_{n\to\infty}\|\delta\|_F = 0$, and with probability 1, $\lambda_{c-1}(\frac{1}{4}\overline{\boldsymbol{W}}^\mathsf{T}\widetilde{\mathbf{A}}\overline{\boldsymbol{W}}) \geqslant \eta$ which is independent of n, we have with probability 1,

$$\lim_{n\to\infty}\left\|\sin\Theta(S_1, S_2)\right\|_F = 0, \tag{C.223}$$

which indicates that the top $c-1$ eigenspaces for $\boldsymbol{M}$ and $\frac{1}{4}\overline{\boldsymbol{W}}^\mathsf{T}\widetilde{\mathbf{A}}\overline{\boldsymbol{W}}$ are the same when $n\to\infty$.

Here we note that the top $c-1$ eigenspace of $\overline{\boldsymbol{W}}^\mathsf{T}\widetilde{\mathbf{A}}\overline{\boldsymbol{W}}$ is $\mathcal{R}\{\overline{\boldsymbol{W}}_i\}_{i=1}^c \backslash \{\mathbf{1}^\mathsf{T}\overline{\boldsymbol{W}}\}$ since $\mathbf{A}$ has its null space spanned by the all-one vector, so $\boldsymbol{M}$ will also have the same top $c-1$ eigenspaces. Besides, from equation Lemma C.64 we know that $\lim_{n\to\infty}\left\|\boldsymbol{W} - \overline{\boldsymbol{W}}\right\|_F = 0$, so $\mathcal{R}\{\overline{\boldsymbol{W}}_i\}_{i=1}^c \backslash \{\mathbf{1}^\mathsf{T}\overline{\boldsymbol{W}}\}$ are the same as $\mathcal{R}\{\boldsymbol{W}_i\}_{i=1}^c \backslash \{\mathbf{1}^\mathsf{T}\boldsymbol{W}\}$. This completes the proof of this theorem.

Structure of Full Hessian of the First Layer

We first restate Theorem 4.4.1 here:

Theorem 4.4.1: *Let V_1 and V_2 be the top $c-1$ eigenspaces of $\boldsymbol{H}$ and $\widehat{\boldsymbol{H}}$ respectively,*

for all $\epsilon > 0$,

$$\lim_{n\to\infty} \Pr_{\boldsymbol{W}^{(1)}\sim\mathcal{N}(0,\frac{1}{d}\boldsymbol{I}_{nd}),\boldsymbol{W}^{(2)}\sim\mathcal{N}(0,\frac{1}{n}\boldsymbol{I}_{cn})} \left[\text{Overlap}\left(V_1, V_2\right) > 1-\epsilon\right] = 1. \tag{C.224}$$

Moreover,

$$\lim_{n\to\infty} \Pr_{\boldsymbol{W}^{(1)}\sim\mathcal{N}(0,\frac{1}{d}\boldsymbol{I}_{nd}),\boldsymbol{W}^{(2)}\sim\mathcal{N}(0,\frac{1}{n}\boldsymbol{I}_{cn})} \left[\left(\frac{\lambda_c(\boldsymbol{H})}{\lambda_{c-1}(\boldsymbol{H})}\bigg|_{\boldsymbol{W}^{(1)},\boldsymbol{W}^{(2)}}\right) < \epsilon\right] = 1. \tag{C.225}$$

Before proceeding to the main theorem, we will first look into the eigenspectrum of the scaled auto-correlation matrix $\breve{\mathbf{X}} \triangleq \frac{1}{d}\mathbb{E}\left[\mathbf{x}\mathbf{x}^\mathsf{T}\right]$ and the top eigenspace of $\widehat{\boldsymbol{H}}$. Also recall some useful notations including $\boldsymbol{U} = \frac{1}{\sqrt{d}}\mathbf{1}_d^\mathsf{T}$ and $\boldsymbol{V} \triangleq \boldsymbol{W} \otimes \boldsymbol{U}$.

Lemma C.2.28. $\lambda_1(\breve{\mathbf{X}}) = \frac{1}{2\pi} + \frac{\pi-1}{2\pi d}$ *with eigenvector* $\frac{1}{\sqrt{d}}\mathbf{1}_d$. $\lambda_2(\breve{\mathbf{X}}) = \cdots = \lambda_d(\breve{\mathbf{X}}) = \frac{\pi-1}{2\pi d}$.

Proof of Lemma C.2.28. From Equation C.165 we know that

$$\breve{\mathbf{X}} = \frac{1}{d}\mathbb{E}\left[\mathbf{x}\mathbf{x}^\mathsf{T}\right] = \frac{1}{2\pi d}\mathbf{1}_d\mathbf{1}_d^\mathsf{T} + \frac{\pi-1}{2\pi d}\boldsymbol{I}_d. \tag{C.226}$$

For unit vector $\boldsymbol{v} = \frac{1}{\sqrt{d}}\mathbf{1}_d$, it satisfies

$$\breve{\mathbf{X}}\boldsymbol{v} = \frac{1}{2\pi d}\mathbf{1}_d\mathbf{1}_d^\mathsf{T}\frac{1}{\sqrt{d}}\mathbf{1}_d + \frac{\pi-1}{2\pi d}\boldsymbol{I}_d\frac{1}{\sqrt{d}}\mathbf{1}_d = \left(\frac{1}{2\pi} + \frac{\pi-1}{2\pi d}\right)\frac{1}{\sqrt{d}}\mathbf{1}_d. \tag{C.227}$$

Hence the all one vector has eigenvalue $\frac{1}{2\pi} + \frac{\pi-1}{2\pi d}$. For any unit vector $\boldsymbol{v} \perp \mathbf{1}_d$, it satisfies

$$\breve{\mathbf{X}}\boldsymbol{v} = \frac{1}{2\pi d}\mathbf{1}_d\mathbf{1}_d^\mathsf{T}\boldsymbol{v} + \frac{\pi-1}{2\pi d}\boldsymbol{I}_d\boldsymbol{v} = \frac{\pi-1}{2\pi d}\boldsymbol{v}. \tag{C.228}$$

Which means $\lambda_2 = \lambda_3 = \cdots = \lambda_d = \frac{\pi-1}{2\pi d}$. □

Corollary C.2.29. *With probability 1 over $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$, the overlap between the top $c-1$ eigenspace of $\widehat{\boldsymbol{H}}$ and $\mathcal{R}\{\boldsymbol{V}_i\}_{i=1}^c \backslash \{\boldsymbol{V}\cdot\boldsymbol{1}\}$ converges to 1 as $n\to\infty$.*

Proof of Corollary C.2.29. First note that by simple linear algebra,

$$\mathcal{R}\{\boldsymbol{V}_i\}_{i=1}^c \backslash \{\boldsymbol{1}^\mathsf{T}\boldsymbol{V}\} = (\mathcal{R}\{\boldsymbol{W}_i\}_{i=1}^c \backslash \{\boldsymbol{W}\cdot\boldsymbol{1}\}) \otimes \boldsymbol{U}. \tag{C.229}$$

While from Theorem 4.4.2 we know the overlap between the top $c-1$ eigenspace of $\boldsymbol{M}$ and $\mathcal{R}\{\boldsymbol{W}_i\}_{i=1}^c \backslash \{\boldsymbol{1}^\mathsf{T}\boldsymbol{W}\}$ converges to 1. Thus for proving this corollary it is sufficient to show that the top $c-1$ eigenspace of $\frac{1}{d}\widehat{\boldsymbol{H}} = \boldsymbol{M}\otimes\breve{\mathbf{X}}$ is the Kronecker product of the top $c-1$ eigenspace of $\boldsymbol{M}$ and $\boldsymbol{U}$.

From Theorem 4.4.2 we know, with probability 1 over $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$, for large n, $\lambda_{c-1}(\boldsymbol{M}) > \eta/4$ and $\lambda_1(\boldsymbol{M}) < 2c^2$ where η and c are absolute constants. Thus for large n we have

$$\lim_{n\to\infty} \lambda_1(\breve{\mathbf{X}})\lambda_{c-1}(\boldsymbol{M}) = \lim_{d\to\infty}\left(\frac{1}{2\pi} + \frac{\pi-1}{2\pi d}\right)\lambda_{c-1}(\boldsymbol{M}) \geqslant \frac{1}{2\pi}\frac{\eta}{4}. \tag{C.230}$$

while

$$\lim_{n\to\infty} \lambda_2(\breve{\mathbf{X}})\lambda_1(\boldsymbol{M}) = \lim_{d\to\infty} \frac{\pi-1}{2\pi d}\lambda_{c-1}(\boldsymbol{M}) \leqslant \lim_{d\to\infty}\frac{\pi-1}{2\pi d}2c^2 = 0. \tag{C.231}$$

Since for large n, $\lambda_1(\breve{\mathbf{X}})\lambda_{c-1}(\boldsymbol{M}) > \lambda_2(\breve{\mathbf{X}})\lambda_1(\boldsymbol{M})$, the top $c-1$ eigenspace of $\frac{1}{d}\widehat{\boldsymbol{H}}$ is the top $c-1$ eigenspace of $\boldsymbol{M}$ Kronecker with the first eigenvector of $\breve{\mathbf{X}}$, which is exactly $\boldsymbol{U}$ from Lemma C.2.28. This completes the proof of this corollary.

□

Now we proceed to prove the main theorem

Proof of Theorem 4.4.1. We will conduct the proof on $\breve{\boldsymbol{H}} = \frac{1}{d}\boldsymbol{H}$ as the properties to be proved are invariant to scalar multiplication. From Corollary C.2.29 we know

the overlap between the top $c-1$ eigenspace of $\widehat{\boldsymbol{H}}$ and $\mathcal{R}\{\boldsymbol{V}_i\}_{i=1}^c \backslash \{\mathbf{1}^\mathsf{T}\boldsymbol{V}\}$ converges to 1. Thus we only need to show the overlap between the top $c-1$ eigenspace of $\breve{\boldsymbol{H}}$ and $\mathcal{R}\{\boldsymbol{V}_i\}_{i=1}^c \backslash \{\mathbf{1}^\mathsf{T}\boldsymbol{V}\}$ converges to 1.

The proof strategy for the full layerwise Hessian is exactly the same as the proof for the output Hessian in Section C.2.2. In particular, the proof is nearly identical when we change the projection matrix from $P_{\boldsymbol{W}}$ to $P_{\boldsymbol{V}}$ where $\boldsymbol{V} \triangleq \boldsymbol{W} \otimes \boldsymbol{U}$.

Therefore, instead of rewriting the entire proof, we may neglect some repeating arguments by verifying the equivalent lemmas for the full layer-wise Hessian. With $\boldsymbol{V}$ as defined, we have $\|\boldsymbol{V}\|_F = \|\boldsymbol{W}\|_F$, $\boldsymbol{V}\boldsymbol{V}^\mathsf{T} = \boldsymbol{W}\boldsymbol{W}^\mathsf{T}$, and $\left\|\boldsymbol{V}^\mathsf{T}\boldsymbol{V}\right\|_F = \left\|\boldsymbol{W}^\mathsf{T}\boldsymbol{W}\right\|_F$, so we can directly apply the exact same result of the two norm bounds (Lemma C.2.3, Lemma C.2.7) on $\boldsymbol{V}$. Now we prove Lemma C.2.30 as the equivalent of Lemma C.2.8.

Lemma C.2.30. *Let $\overline{\boldsymbol{V}} \triangleq \overline{\boldsymbol{W}} \otimes \boldsymbol{U}$, then $P_{\boldsymbol{V}} \triangleq \overline{\boldsymbol{V}}^\mathsf{T}\overline{\boldsymbol{V}}$ is the projection matrix from $\mathbb{R}^{nd}$ onto the subspace spanned by all rows of $\boldsymbol{V} = \boldsymbol{W} \otimes \boldsymbol{U}$. Moreover, for all $\epsilon > 0$,*

$$\lim_{n\to\infty} \Pr\left[\left\|\boldsymbol{V}^\mathsf{T}\boldsymbol{V} - P_{\boldsymbol{V}}\right\|_F^2 > \epsilon\right] = 0. \tag{C.232}$$

Proof of Lemma C.2.30. Since Kronecker product with the constant $1 \times d$ matrix $\boldsymbol{U}$ preserves the orthogonality of vectors, doing Gram-Schmit on $\boldsymbol{V}$ is equivalent to doing Gram-Schmit on $\boldsymbol{W}$ then Kronecker with $\boldsymbol{U}$, which results in $\overline{\boldsymbol{V}}$ by construction. Therefore $P_{\boldsymbol{V}}$ is a valid projection matrix.

Also note that for any $\boldsymbol{W}$,

$$\begin{aligned}
\left\|\boldsymbol{V}^{\mathsf{T}}\boldsymbol{V} - P_{\boldsymbol{V}}\right\|_F^2 &= \left\|(\boldsymbol{W}\otimes\boldsymbol{U})^{\mathsf{T}}(\boldsymbol{W}\otimes\boldsymbol{U}) - \overline{\boldsymbol{V}}^{\mathsf{T}}\overline{\boldsymbol{V}}\right\|_F^2 \\
&= \left\|(\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W})\otimes(\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}) - (\overline{\boldsymbol{W}}^{\mathsf{T}}\overline{\boldsymbol{W}})\otimes(\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U})\right\|_F^2 \\
&= \left\|\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W} - \overline{\boldsymbol{W}}^{\mathsf{T}}\overline{\boldsymbol{W}}\right\|_F^2 \left\|\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}\right\|_F^2 \qquad \text{(C.233)} \\
&= \left\|\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W} - \overline{\boldsymbol{W}}^{\mathsf{T}}\overline{\boldsymbol{W}}\right\|_F^2 \left\|\frac{1}{d}\mathbf{1}_d\mathbf{1}_d^{\mathsf{T}}\right\|_F^2 \\
&= \left\|\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W} - P_{\boldsymbol{W}}\right\|_F^2.
\end{aligned}$$

From Lemma C.2.8 we have

$$\lim_{n\to\infty}\Pr\left[\left\|\boldsymbol{V}^{\mathsf{T}}\boldsymbol{V} - P_{\boldsymbol{V}}\right\|_F^2 > \epsilon\right] = \lim_{n\to\infty}\Pr\left[\left\|\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W} - P_{\boldsymbol{W}}\right\|_F^2 > \epsilon\right] = 0. \qquad \text{(C.234)}$$

□

Note that the equivalent lemmas of Lemma C.2.18- Lemma C.2.22 for $\breve{\boldsymbol{H}}$ are also established in Lemma C.2.23 - Lemma C.2.27 in Section C.2.2, substituting $(P_{\boldsymbol{W}}, \boldsymbol{M}, \boldsymbol{M}^*)$ by $(P_{\boldsymbol{V}}, \breve{\boldsymbol{H}}, \breve{\boldsymbol{H}}^*)$, we may follow the argument in Section C.2.2 up to Equation C.208 and conclude that

$$\lim_{n\to\infty}\left\|\breve{\boldsymbol{H}} - \boldsymbol{V}^{\mathsf{T}}\boldsymbol{V}\breve{\boldsymbol{H}}^*\boldsymbol{V}^{\mathsf{T}}\boldsymbol{V}\right\|_F = 0. \qquad \text{(C.235)}$$

Now claim an equivalent argument of Equation C.216, that

$$\lim_{n\to\infty}\left\|\frac{1}{8\pi}\boldsymbol{V}^{\mathsf{T}}\tilde{\mathbf{A}}\boldsymbol{V} - \boldsymbol{V}^{\mathsf{T}}\boldsymbol{V}\breve{\boldsymbol{H}}^*\boldsymbol{V}^{\mathsf{T}}\boldsymbol{V}\right\|_F = 0. \qquad \text{(C.236)}$$

Observe that

$$\boldsymbol{V}^{\mathsf{T}}\tilde{\mathbf{A}}\boldsymbol{V} = (\boldsymbol{W}\otimes\boldsymbol{U})^{\mathsf{T}}\tilde{\mathbf{A}}(\boldsymbol{W}\otimes\boldsymbol{U}) = \boldsymbol{W}^{\mathsf{T}}\tilde{\mathbf{A}}\boldsymbol{W}\otimes\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U} \qquad \text{(C.237)}$$

and

$$\begin{aligned}\boldsymbol{V}^{\mathsf{T}}\boldsymbol{V}\breve{\boldsymbol{H}}^{*}\boldsymbol{V}^{\mathsf{T}}\boldsymbol{V} &= (\boldsymbol{W}\otimes\boldsymbol{U})^{\mathsf{T}}(\boldsymbol{W}\otimes\boldsymbol{U})(\boldsymbol{M}^{*}\otimes\breve{\mathbf{X}})(\boldsymbol{W}\otimes\boldsymbol{U})^{\mathsf{T}}(\boldsymbol{W}\otimes\boldsymbol{U}) \\ &= \boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\otimes\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}\breve{\mathbf{X}}\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}.\end{aligned} \tag{C.238}$$

We have

$$\begin{aligned}&\left\|\frac{1}{8\pi}\boldsymbol{V}^{\mathsf{T}}\tilde{\mathbf{A}}\boldsymbol{V}-\boldsymbol{V}^{\mathsf{T}}\boldsymbol{V}\breve{\boldsymbol{H}}^{*}\boldsymbol{V}^{\mathsf{T}}\boldsymbol{V}\right\|_{F} \\ =&\left\|\frac{1}{8\pi}\boldsymbol{W}^{\mathsf{T}}\tilde{\mathbf{A}}\boldsymbol{W}\otimes\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}-\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\otimes\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}\breve{\mathbf{X}}\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}\right\|_{F} \\ \leqslant&\left\|\frac{1}{8\pi}\boldsymbol{W}^{\mathsf{T}}\tilde{\mathbf{A}}\boldsymbol{W}\otimes\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}-\frac{1}{2\pi}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\otimes\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}\right\|_{F} \\ &+\left\|\frac{1}{2\pi}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\otimes\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}-\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\otimes\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}\breve{\mathbf{X}}\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}\right\|_{F} \\ =&\frac{1}{2\pi}\left\|\frac{1}{4}\boldsymbol{W}^{\mathsf{T}}\tilde{\mathbf{A}}\boldsymbol{W}-\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_{F}\left\|\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}\breve{\mathbf{X}}\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}\right\|_{F} \\ &+\left\|\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_{F}\left\|\frac{1}{2\pi}\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}-\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}\breve{\mathbf{X}}\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}\right\|_{F}.\end{aligned} \tag{C.239}$$

Let's first consider the second term. Note that from Lemma C.2.28,

$$\begin{aligned}\left\|\frac{1}{2\pi}\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}-\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}\breve{\mathbf{X}}\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}\right\|_{F} &= \left\|\frac{1}{2\pi d}\mathbf{1}_{d}\mathbf{1}_{d}^{\mathsf{T}}-\frac{1}{d^{2}}\left(\sum_{i,j=1}^{d}\frac{1}{d}\mathbb{E}\left[\mathbf{x}\mathbf{x}^{\mathsf{T}}\right]_{ij}\right)\mathbf{1}_{d}\mathbf{1}_{d}^{\mathsf{T}}\right\|_{F} \\ &= \left|\frac{1}{2\pi d}-\frac{1}{d^{3}}\left(\frac{2\pi d^{2}}{2\pi}+\frac{(\pi-1)d}{2\pi}\right)\right|\left\|\mathbf{1}_{d}\mathbf{1}_{d}^{\mathsf{T}}\right\|_{F} \\ &= \frac{\pi-1}{2\pi d^{2}}d=\frac{\pi-1}{2\pi d}.\end{aligned} \tag{C.240}$$

Which converges to 0 as $n\to\infty$ (since $d=n^{1+\alpha}$). Since $\left\|\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{M}^{*}\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W}\right\|_{F}$ is

bounded above from Lemma C.2.3 and Lemma C.2.19. We have

$$\lim_{n\to\infty} \left\|\boldsymbol{W}^\mathsf{T}\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\mathsf{T}\boldsymbol{W}\right\|_F \left\|\frac{1}{2\pi}\boldsymbol{U}^\mathsf{T}\boldsymbol{U} - \boldsymbol{U}^\mathsf{T}\boldsymbol{U}\breve{\mathbf{X}}\boldsymbol{U}^\mathsf{T}\boldsymbol{U}\right\|_F = 0. \tag{C.241}$$

For the first term, since for all d,

$$\begin{aligned}\left\|\boldsymbol{U}^\mathsf{T}\boldsymbol{U}\breve{\mathbf{X}}\boldsymbol{U}^\mathsf{T}\boldsymbol{U}\right\|_F &= \left\|\frac{1}{d^2}\left(\sum_{i,j=1}^{d}\frac{1}{d}\mathbb{E}\left[\mathbf{x}\mathbf{x}^\mathsf{T}\right]_{ij}\right)\mathbf{1}_d\mathbf{1}_d^\mathsf{T}\right\|_F \\ &= \frac{1}{d^2}\left(\frac{2\pi d^2}{2\pi} + \frac{(\pi-1)d}{2\pi}\right) < \frac{1}{2},\end{aligned} \tag{C.242}$$

Combined with Equation C.216 we have

$$\lim_{n\to\infty}\frac{1}{2\pi}\left\|\frac{1}{4}\boldsymbol{W}^\mathsf{T}\tilde{\mathbf{A}}\boldsymbol{W} - \boldsymbol{W}^\mathsf{T}\boldsymbol{W}\boldsymbol{M}^*\boldsymbol{W}^\mathsf{T}\boldsymbol{W}\right\|_F\left\|\boldsymbol{U}^\mathsf{T}\boldsymbol{U}\breve{\mathbf{X}}\boldsymbol{U}^\mathsf{T}\boldsymbol{U}\right\|_F = 0. \tag{C.243}$$

Plug Equation C.241 and Equation C.243 into Equation C.239 gives us Equation C.236.

Now substitute $\frac{1}{4}\boldsymbol{W}^\mathsf{T}\boldsymbol{A}\boldsymbol{W}$ in Section C.2.2 to $\frac{1}{8\pi}\boldsymbol{V}^\mathsf{T}\boldsymbol{A}\boldsymbol{V}$, following the arguments after Equation C.216 completes the remaining proof for this theorem. □

C.3 Structure of Dominating Eigenvectors of the Full Hessian.

Although it is not possible to apply Kronecker factorization to the full Hessian directly, we can construct an approximation of the top eigenvectors and eigenspace using similar ideas and our findings.In this section, we will always have superscript (p) for all layer-wise matrices and vectors in order to distinguish them from the full versions. As shown in Equation C.19 of Section C.1.1, we have the full Hessian of fully connected networks as

$$\boldsymbol{H}_{\mathcal{L}}(\theta) = \mathbb{E}\left[\boldsymbol{F}_{\boldsymbol{x}}^{\mathsf{T}} \boldsymbol{A}_{\boldsymbol{x}} \boldsymbol{F}_{\boldsymbol{x}}\right] + \mathbb{E}\left[\sum_{i=1}^{c} \frac{\partial \ell(\boldsymbol{z}, \boldsymbol{y})}{z_i} \nabla_{\theta}^2 z_i\right], \tag{C.244}$$

where

$$\boldsymbol{F}_{\boldsymbol{x}}^{\mathsf{T}} = \begin{pmatrix} {\boldsymbol{G}_{\boldsymbol{x}}}^{(1)\mathsf{T}} \otimes \boldsymbol{x}^{(1)} \\ {\boldsymbol{G}_{\boldsymbol{x}}}^{(1)\mathsf{T}} \\ {\boldsymbol{G}_{\boldsymbol{x}}}^{(2)\mathsf{T}} \otimes \boldsymbol{x}^{(2)} \\ {\boldsymbol{G}_{\boldsymbol{x}}}^{(2)\mathsf{T}} \\ \vdots \\ {\boldsymbol{G}_{\boldsymbol{x}}}^{(L)\mathsf{T}} \otimes \boldsymbol{x}^{(n)} \\ {\boldsymbol{G}_{\boldsymbol{x}}}^{(L)\mathsf{T}} \end{pmatrix}. \tag{C.245}$$

In order to simplify the formula, we define

$$\widetilde{\boldsymbol{x}}^{(p)} = \begin{pmatrix} \boldsymbol{x}^{(p)} \\ 1 \end{pmatrix} \tag{C.246}$$

to be the extended input of the p-th layer. Thus, the terms in the Hessian attributed to the bias can be included in the Kronecker product with the extended input, and

$\boldsymbol{F}_{\boldsymbol{x}}^{\mathsf{T}}$ can be simplified as

$$\boldsymbol{F}_{\boldsymbol{x}}^{\mathsf{T}} = \begin{pmatrix} \boldsymbol{G}_{\boldsymbol{x}}{}^{(1)\mathsf{T}} \otimes \widetilde{\boldsymbol{x}}^{(1)} \\ \boldsymbol{G}_{\boldsymbol{x}}{}^{(2)\mathsf{T}} \otimes \widetilde{\boldsymbol{x}}^{(2)} \\ \vdots \\ \boldsymbol{G}_{\boldsymbol{x}}{}^{(L)\mathsf{T}} \otimes \widetilde{\boldsymbol{x}}^{(n)} \end{pmatrix}. \tag{C.247}$$

As discussed in several previous works (Sagun et al., 2016; Papyan, 2018, 2019; Fort and Ganguli, 2019), the full Hessian can be decomposed in to the G-term $\mathbb{E}\left[\boldsymbol{F}_{\boldsymbol{x}}^{\mathsf{T}}\boldsymbol{A}_{\boldsymbol{x}}\boldsymbol{F}_{\boldsymbol{x}}\right]$ and the H-term $\mathbb{E}\left[\sum_{i=1}^{c}\frac{\partial \ell(\boldsymbol{z},\boldsymbol{y})}{z_i}\nabla_\theta^2 z_i\right]$ in Equation C.244.

Empirically, the G-term usually dominates the H-term, and the top eigenvalues and eigenspace of the Hessian are mainly attributed to the G-term. Since we focus on the top eigenspace, we can approximate our full Hessian using the G-term, as

$$\boldsymbol{H}_{\mathcal{L}}(\theta) \approx \mathbb{E}\left[\boldsymbol{F}_{\boldsymbol{x}}^{\mathsf{T}}\boldsymbol{A}_{\boldsymbol{x}}\boldsymbol{F}_{\boldsymbol{x}}\right]. \tag{C.248}$$

In our approximation of the layer-wise Hessian $\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)})$ Equation 4.2, the two parts of the Kronecker factorization are the layer-wise output Hessian $\mathbb{E}[\boldsymbol{M}_{\boldsymbol{x}}^{(p)}]$ and the auto-correlation matrix of the input $\mathbb{E}[\boldsymbol{x}^{(p)}\boldsymbol{x}^{(p)\mathsf{T}}]$. Although we cannot apply Kronecker factorization to $\mathbb{E}\left[\boldsymbol{F}_{\boldsymbol{x}}^{\mathsf{T}}\boldsymbol{A}_{\boldsymbol{x}}\boldsymbol{F}_{\boldsymbol{x}}\right]$, we can still approximate its eigenspace using the eigenspace of the full output Hessian.

Note here that the full output Hessian is not a common definition. Let $\widehat{m} = \sum_{p=1}^{L} m^{(p)}$ be the sum of output dimension of each layer. We define a full output vector $\widetilde{\boldsymbol{z}} \in \mathbb{R}^{\widehat{m}}$ by concatenating all the layerwise outputs together,

$$\widetilde{\boldsymbol{z}} := \begin{pmatrix} \boldsymbol{z}^{(1)} \\ \boldsymbol{z}^{(2)} \\ \vdots \\ \boldsymbol{z}^{(L)} \end{pmatrix}. \tag{C.249}$$

We then define the full output Hessian is the Hessian w.r.t. $\widetilde{\boldsymbol{z}}$. Let the full output Hessian for a single input $\boldsymbol{x}$ be $\boldsymbol{M}_{\boldsymbol{x}} \in \mathbb{R}^{\widehat{m} \times \widehat{m}}$. Similar to Equation C.15, it can be expressed as

$$\boldsymbol{M}_{\boldsymbol{x}} := \boldsymbol{H}_{\ell}(\widetilde{\boldsymbol{z}}, \boldsymbol{x}) = \boldsymbol{G}_{\boldsymbol{x}}^{\mathsf{T}} \boldsymbol{A}_{\boldsymbol{x}} \boldsymbol{G}_{\boldsymbol{x}}, \tag{C.250}$$

where

$$\boldsymbol{G}_{\boldsymbol{x}}^{\mathsf{T}} = \begin{pmatrix} \boldsymbol{G}_{\boldsymbol{x}}^{(1)\mathsf{T}} \\ \boldsymbol{G}_{\boldsymbol{x}}^{(2)\mathsf{T}} \\ \vdots \\ \boldsymbol{G}_{\boldsymbol{x}}^{(L)\mathsf{T}} \end{pmatrix} \tag{C.251}$$

similar to Equation C.247. The full output Hessian for the entire training sample is thus

$$\boldsymbol{H}_{\mathcal{L}}(\widetilde{\boldsymbol{z}}) = \mathbb{E}[\boldsymbol{M}_{\boldsymbol{x}}] = \mathbb{E}[\boldsymbol{G}_{\boldsymbol{x}}^{\mathsf{T}} \boldsymbol{A}_{\boldsymbol{x}} \boldsymbol{G}_{\boldsymbol{x}}]. \tag{C.252}$$

We can then approximate the eigenvectors of the full Hessian $\boldsymbol{H}_{\mathcal{L}}(\theta)$ using the eigenvectors of $\mathbb{E}[\boldsymbol{M}_{\boldsymbol{x}}]$. Let the i-th eigenvector of $\boldsymbol{H}_{\mathcal{L}}(\theta)$ be $\boldsymbol{v}_i$ and that of $\mathbb{E}[\boldsymbol{M}_{\boldsymbol{x}}]$ be $\boldsymbol{u}_i$. We may then break up $\boldsymbol{u}_i$ into segments corresponding to different layers as in

$$\boldsymbol{u}_i = \begin{pmatrix} \boldsymbol{u}_i^{(1)} \\ \boldsymbol{u}_i^{(2)} \\ \vdots \\ \boldsymbol{u}_i^{(L)} \end{pmatrix}, \tag{C.253}$$

where for all layer p, $\boldsymbol{u}_i^{(p)} \in \mathbb{R}^{m^{(p)}}$. Motivated by the relation between $\boldsymbol{G}_{\boldsymbol{x}}$ and $\boldsymbol{F}_{\boldsymbol{x}}$, the i-th eigenvector of $\boldsymbol{H}_{\mathcal{L}}(\theta)$ can be approximated as the following. Let

$$\boldsymbol{w}_i = \begin{pmatrix} \boldsymbol{u}_i^{(1)} \otimes \mathbb{E}[\widetilde{\boldsymbol{x}^{(1)}}] \\ \boldsymbol{u}_i^{(2)} \otimes \mathbb{E}[\widetilde{\boldsymbol{x}^{(2)}}] \\ \vdots \\ \boldsymbol{u}_i^{(L)} \otimes \mathbb{E}[\widetilde{\boldsymbol{x}^{(L)}}] \end{pmatrix}. \tag{C.254}$$

We then have

$$\boldsymbol{v}_i \approx \frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|} \tag{C.255}$$

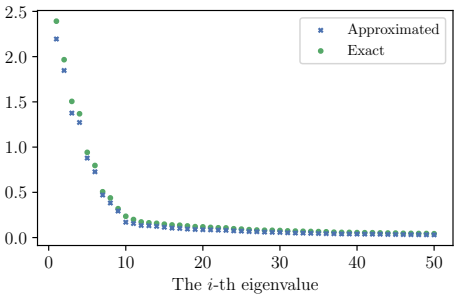
We can then use the Gram–Schmidt process to get the basis vectors of the approximated eigenspace.

Another reason for this approximation is that the expectation is the input of each layer $\mathbb{E}[\boldsymbol{x}^{(p)}]$ dominates its covariance as shown in Section C.6.1. Thus, the approximate is accurate for top eigenvectors and also top eigenspace. For latter eigenvectors, the approximation would not be as accurate since this approximate loses all information in the covariance of the inputs.

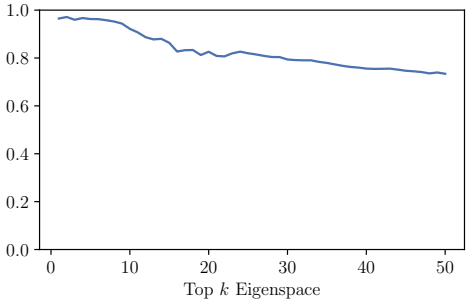
We also approximated the eigenvalues using this approximation. Let the i-th eigenvalue of $\boldsymbol{H}_{\mathcal{L}}(\theta)$ be λ_i and that of $\mathbb{E}[\boldsymbol{M}_{\boldsymbol{x}}]$ be σ_i. We have

$$\lambda_i \approx \sigma_i \|\boldsymbol{w}_i\|^2. \tag{C.256}$$

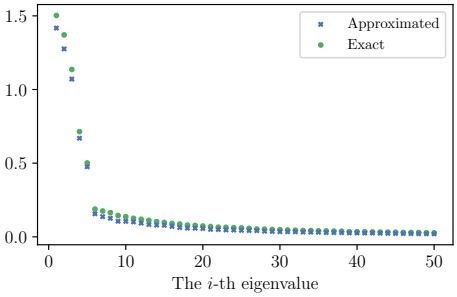
Below we show the approximation of the eigenvalues top eigenspace using this method. The eigenspace overlap is defined as in Definition 4.2.1. We experimented on several fully connected networks, the results shown below are for F-200^2 (same as Figure 4.2(c)(d) in the main text), F-200^4, F-600^4, and F-600^8, all with dimension 50. The approximations are reasonably accurate.



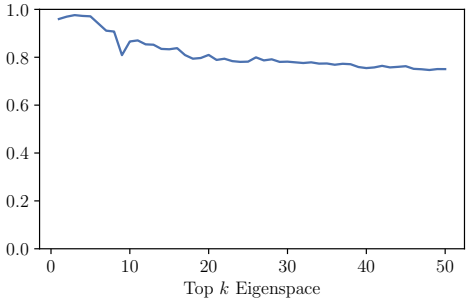
(a) Eigenvalues for F-200^2



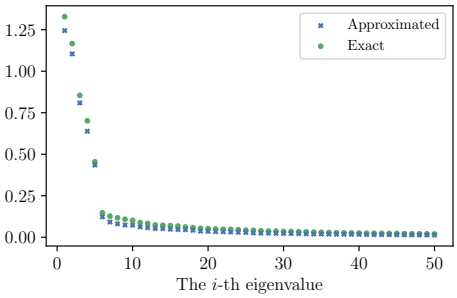
(b) Eigenspace overlap for F-200^2



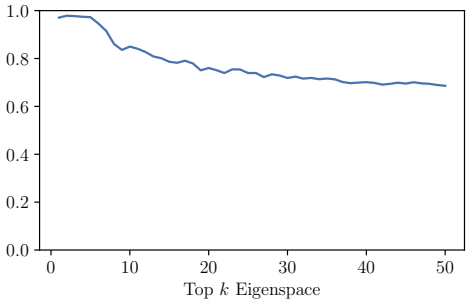
(c) Eigenvalues for F-200^4



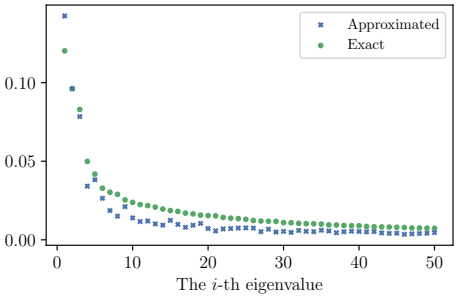
(d) Eigenspace overlap for F-200^4



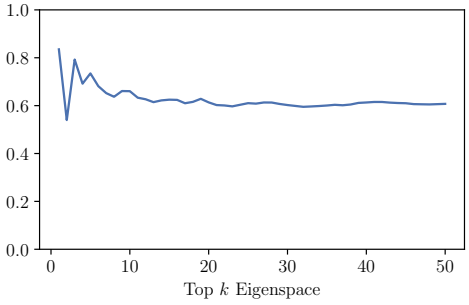
(e) Eigenvalues for F-600^4



(f) Eigenspace overlap for F-600^4



(g) Eigenvalues for F-600^8



(h) Eigenspace overlap for F-600^8

FIGURE C.1: Top 50 Eigenvalues and Eigenspace approximation for full Hessian

C.4 Computation of Hessian Eigenvalues and Eigenvectors

For Hessian approximated using Kronecker factorization, we compute $\mathbb{E}[\boldsymbol{M}]$ and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T]$ explicitly. Let $\boldsymbol{m}$ and $\boldsymbol{v}$ be an eigenvector of $\mathbb{E}[\boldsymbol{M}]$ and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T]$ respectively, with corresponding eigenvalues $\lambda_{\boldsymbol{m}}$ and $\lambda_{\boldsymbol{v}}$. Since both matrices are positive semi-definite, $\boldsymbol{m} \otimes \boldsymbol{v}$ is an eigenvector of $\mathbb{E}[\boldsymbol{M}] \otimes \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T]$ with eigenvalue $\lambda_{\boldsymbol{m}}\lambda_{\boldsymbol{v}}$. In this way, since $\mathbb{E}[\boldsymbol{M}]$ has m eigenvectors and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T]$ has n eigenvectors, we can approximate all mn eigenvectors for the layer-wise Hessian. All these calculation can be done directly.

However, it is almost prohibitive to calculate the true Hessian explicitly. Thus, we use numerical methods with automatic differentiation (Paszke et al., 2017) to calculate them. The packages we use is Golmant et al. (2018) and we use the Lanczos method in most of the calculations. We also use package in Yao et al. (2019) as a reference.

For layer-wise Hessian, we modified the Golmant et al. (2018) package. In particular, the package relies on the calculation of Hessian-vector product $\boldsymbol{H}\boldsymbol{v}$, where $\boldsymbol{v}$ is a vector with the same size as parameter θ. To calculate eigenvalues and eigenvectors for layer-wise Hessian at the p-th layer, we cut the $\boldsymbol{v}$ into different layers. Then, we only leave the part corresponding to weights of the p-th layer and set all other entries to 0. Note that the dimension does not change. We let the new vector be $\boldsymbol{v}^{(p)}$ and get the value of $\boldsymbol{u} = \boldsymbol{H}\boldsymbol{v}^{(p)}$ using auto differentiation. Then, we do the same operation to $\boldsymbol{u}$ and get $\boldsymbol{u}^{(p)}$.

C.5 Detailed Experiment Setup

C.5.1 Datasets

We conduct experiment on CIFAR-10, CIFAR-100 (MIT) (Krizhevsky and Hinton, 2009) (`https://www.cs.toronto.edu/~kriz/cifar.html`), and MNIST (CC BY-SA 3.0) (LeCun et al., 1998) (`http://yann.lecun.com/exdb/mnist/`). The datasets are downloaded through torchvision (Paszke et al., 2019b) (`https://pytorch.org/vision/stable/index.html`). We used their default splitting of training and testing set.

To compare our work on PAC-Bayes bound with the work of Dziugaite and Roy (2017), we created a custom dataset MNIST-2 by setting the label of images 0-4 to 0 and 5-9 to 1. We also created random-labeled datasets MNIST-R and CIFAR10-R by randomly labeling the images from the training set of MNIST and CIFAR10. The dataset information is summarized in Table C.1

Table C.1: Datasets

Dataset	# Data Points		Input Size	# Classes	Label
	Train	Test			
CIFAR10	50000	10000	$3 \times 32 \times 32$	10	True
CIFAR10-R	50000	10000	$3 \times 32 \times 32$	10	Random
CIFAR100	50000	10000	$3 \times 32 \times 32$	100	True
MNIST	60000	10000	28×28	10	True
MNIST-2	60000	10000	28×28	2	True
MNIST-R	60000	10000	28×28	10	Random

All the datasets (MNIST, CIFAR-10, and CIFAR-100) we used are publicly available. According to their descriptions on the contents and collection methods, they should not contain any personal information or offensive content. MNIST is a remix of datasets from the National Institute of Standards and Technology (NIST), which

obtained consent for collecting the data. However, we also note that CIFAR-10 and CIFAR-100 are subsets of the dataset 80 Million Tiny Image (Torralba et al., 2008) (`http://groups.csail.mit.edu/vision/TinyImages/`), which used automatic collection and includes some offensive images.

C.5.2 Network Structures

Fully Connected Network: We used several different fully connected networks varying in the number of hidden layers and the number of neurons for each hidden layer. The output of all layers except the last layer are passed into ReLU before feeding into the subsequent layer. As described in Section 4.5.1, we denote a fully connected network with m hidden layers and n neurons each hidden layer by F-n^m. For networks without uniform layer width, we denote them by a sequence of numbers (e.g. for a network with three hidden layers, where the first two layers has 200 neurons each and the third has 100 neurons, we denote it as F-200^2-100). For example, the structure of F-200^2 is shown in Table C.2.

Table C.2: Structure of F-200^2 on MNIST

#	Name	Module	In Shape	Out Shape
1		Flatten	(28,28)	784
2	fc1	Linear(784, 200)	784	200
3		ReLU	200	200
4	fc2	Linear(200, 200)	200	200
5		ReLU	200	200
6	fc3	Linear(200, 10)	200	10
		output		

LeNet5: We adopted the LeNet5 structure proposed by LeCun et al. (1998) for MNIST, and slightly modified the input convolutional layers to adapt the input of CIFAR-10 dataset. The standard LeNet5 structure we used in the experiments is

shown in Table C.3. We further modified the dimension of fc1 and conv2 to create several variants for the experiment in Section 4.5.3. Take the model whose first fully connected layer is adjusted to have 80 neurons as an example, we denote it as LeNet5-(fc1-80).

Table C.3: Structure of LeNet5 on CIFAR-10

#	Name	Module	In Shape	Out Shape
1	conv1	Conv2D(3, 6, 5, 5)	(3, 32, 32)	(6, 28, 28)
2		ReLU	(6, 28, 28)	(6, 28, 28)
3	maxpool1	MaxPooling2D(2,2)	(6, 28, 28)	(6, 14, 14)
4	conv2	Conv2D(6, 16, 5, 5)	(6, 14, 14)	(16, 10, 10)
5		ReLU	(16, 10, 10)	(16, 10, 10)
6	maxpool2	MaxPooling2D(2,2)	(16, 10, 10)	(16, 5, 5)
7		Flatten	(16, 5, 5)	400
8	fc1	Linear(400, 120)	400	120
9		ReLU	120	120
10	fc2	Linear(120, 84)	120	84
11		ReLU	84	84
12	fc3	Linear(84, 10)	84	10
		output		

Networks with Batch Normalization: In Section C.7.3 we conducted several experiments regarding the effect of batch normalization on our results. For those experiments, we use the existing structures and add batch normalization layer for each intermediate output after it passes the ReLU module. In order for the Hessian to be well-defined, we fix the running statistics of batch normalization and treat it as a linear layer during inference. We also turn off the learnable parameters θ and β (Ioffe and Szegedy, 2015) for simplicity. For network structure X, we denote the variant with batch normalization after all hidden layers X-BN. For example, the detailed structure LeNet5-BN is shown in Table C.4.

Table C.4: Structure of LeNet5-BN on CIFAR-10

#	Name	Module	In Shape	Out Shape
1	conv1	Conv2D(3, 6, 5, 5)	(3, 32, 32)	(6, 28, 28)
2		ReLU	(6, 28, 28)	(6, 28, 28)
3		BatchNorm2D	(6, 28, 28)	(6, 28, 28)
4	maxpool1	MaxPooling2D(2,2)	(6, 28, 28)	(6, 14, 14)
5	conv2	Conv2D(6, 16, 5, 5)	(6, 14, 14)	(16, 10, 10)
6		ReLU	(16, 10, 10)	(16, 10, 10)
7		BatchNorm2D	(16, 10, 10)	(16, 10, 10)
8	maxpool2	MaxPooling2D(2,2)	(16, 10, 10)	(16, 5, 5)
9		Flatten	(16, 5, 5)	400
10	fc1	Linear(400, 120)	400	120
11		ReLU	120	120
12		BatchNorm1D	120	120
13	fc2	Linear(120, 84)	120	84
14		ReLU	84	84
15		BatchNorm1D	84	84
16	fc3	Linear(84, 10)	84	10
		output		

Variants of VGG11: To verify that our results apply to larger networks, we trained a number of variant of VGG11 (originally named VGG-A in the paper, but commonly refered as VGG11) proposed by Simonyan and Zisserman (2015). For simplicity, we removed the dropout regularization in the original network. To adapt the structure, which is originally designed for the $3 \times 224 \times 224$ input of ImageNet, to $3 \times 32 \times 32$ input of CIFAR-10.

Since the original VGG11 network is too large for computing the top eigenspace up to hundreds of dimensions, we reduce the number of output channels of each convolution layer in the network to 32, 48, 64, 80, and 200. We denote the small size variants as VGG11-W32, VGG11-W48, VGG11-W64, VGG11-W80, and VGG11-W200 respectively. We use conv1 - conv8 and fc1 to denote the layers of VGG11 where conv1 is closest to the input feature and fc1 is the classification layer.

Variants of ResNet18: We also trained a number of variant of ResNet18 proposed by He et al. (2016b). As batch normalization will change the low rank structure of the auto correlation matrix and reduce the overlap, we removed all batch normalization operations. Following the adaptation of ResNet to CIFAR dataset as in `https://github.com/kuangliu/pytorch-cifar`, we changed the input size to $3 \times 32 \times 32$ and added a 1x1 convolutional layer for each shortcut after the first block.

Similar to VGG11, we reduce the number of output channels of each convolution layer in the network to 48, 64, 80. We denote the small size variants as ResNet18-W48, ResNet18-W64, and ResNet18-W80 respectively. We use conv1 - conv17 and fc1 to denote the layers of the ResNet18 backbone where conv1 is closest to the input feature and fc1 is the classification layer. For the 1x1 convolutional layers in the shortcut, we denote them by sc-conv1 - sc-conv3. where sc-conv1 is the convolutional layer on the shortcut of the second ResNet block and sc-conv3 is the convolutional layer on the shortcut of the fourth ResNet block.

C.5.3 Training Process and Hyperparameter Configuration

For all datasets, we used the default splitting of training and testing set. All models (except explicitly stated otherwise) are trained using batched stochastic gradient descent (SGD) with batch-size 128 and fixed learning rate 0.01 for 1000 epochs. No momentum and weight decay regularization were used. The loss objective converges by the end of training, so we may assume that the final models are at local minima. For generality we also used a training scheme with fixed learning rate at 0.001, and a training scheme with fixed learning rate at 0.01 with momentum of 0.9 and weight-decay factor of 0.0005. Models trained with these settings will be explicitly stated. Otherwise we assume they were trained with the default scheme mentioned above.

Follow the default initialization scheme of PyTorch(Paszke et al., 2019b), the weights of linear layers and convolutional layers are initialized using the Xavier method (Glorot and Bengio, 2010), and bias of each layer are initialized to be zero.

C.6 Additional Empirical Results

C.6.1 Low Rank Structure of Auto-Correlation Matrix $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$

We have briefly discussed about the autocorrelation matrix $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ being approximately rank 1 in Section 4.3.1 in the main text. In particular, we claimed that the mean of layer input dominate the covariance, that $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}] \approx \mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{x}^\mathsf{T}]$. In this section we provide some additional empirical results supporting that claim.

We use two metrics to quantify the quality of this approximation: the squared dot product between normalized $\mathbb{E}[\boldsymbol{x}]$ and the first eigenvector of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ and the ratio between the first and second eigenvalue of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$. Intuitively if the first quantity is close to 1 and the second quantity is large, then the approximation is accurate. Formally, for fully connected layers, define $\widehat{\mathbb{E}}[\boldsymbol{x}]$ as the normalized expectation of the layer input $\boldsymbol{x}$, namely $\mathbb{E}[\boldsymbol{x}]/\|\mathbb{E}[\boldsymbol{x}]\|$. For convolutional layers, following the notations in Section C.1.2, define $\widehat{\mathbb{E}}[\boldsymbol{x}]$ as the first left singular vector of $\mathbb{E}[\boldsymbol{X}]$ where $\widehat{\mathbb{E}}[\boldsymbol{x}] \in \mathbb{R}^{nK_1K_2}$. Abusing notations for simplicity, we use $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ to denote the $nK_1K_2 \times nK_1K_2$ matrix $\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\mathsf{T}]$. In this section we consider the squared dot product between $\widehat{\mathbb{E}}[\boldsymbol{x}]$ and the first eigenvector $\boldsymbol{v}_1$ of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$, namely $(\boldsymbol{v}_1^\mathsf{T}\widehat{\mathbb{E}}[\boldsymbol{x}])^2$.

For the spectral ratio, let λ_1 be the first eigenvalue of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ and λ_2 be the second. We have

$$\frac{\lambda_1}{\lambda_2} \geqslant \frac{\|\mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{x}]^\mathsf{T}\| - \|\boldsymbol{\Sigma}_{\boldsymbol{x}}\|}{\|\boldsymbol{\Sigma}_{\boldsymbol{x}}\|} = \frac{\|\mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{x}]^\mathsf{T}\|}{\|\boldsymbol{\Sigma}_{\boldsymbol{x}}\|} - 1, \tag{C.257}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ is the covariance of $\boldsymbol{x}$. Thus, the spectral norm of $\mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{x}]^\mathsf{T}$ divided by that of $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ gives a lower bound to λ_1/λ_2. In our experiments, we usually have $\lambda_1/\lambda_2 \geqslant \|\mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{x}]^\mathsf{T}\|/\|\boldsymbol{\Sigma}_{\boldsymbol{x}}\|$.

As we can see from Table C.5 and Table C.6, in a variety of settings, $\mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{x}]^\mathsf{T}$

indeed dominated the autocorrelation matrix $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ for fully connected layers. Similar phenomenon also holds for convolutional layers in the modern architectures, but the spectral gap are generally smaller compared to that of the fully connected layers.

Table C.5: Squared dot product $(\boldsymbol{v}_1^\mathsf{T}\widehat{\mathbb{E}}[\boldsymbol{x}])^2$ and spectral ratio λ_1/λ_2 for fully connected layers in a selection of network structures and datasets. We independently trained 5 runs for each instance and compute the mean, minimum, and maximum of the two quantities over all layers (except the first layer which takes the input with mean-zero) in all runs.

			$(\boldsymbol{v}_1^\mathsf{T}\widehat{\mathbb{E}}[\boldsymbol{x}])^2$			λ_1/λ_2		
Dataset	Network	# fc	mean	min	max	mean	min	max
MNIST	F-200^2	2	1.000	1.000	1.000	12.29	9.65	16.16
	F-600^2	2	0.999	0.999	0.999	12.00	11.42	13.00
	F-600^4	4	1.000	0.999	1.000	17.81	7.33	28.00
	F-600^8	8	0.991	0.965	1.000	6.63	2.28	11.15
CIFAR10	F-600^2	2	0.999	0.998	1.000	9.24	4.74	13.74
	F-1500^3	3	0.999	0.997	1.000	13.27	6.10	18.41
	LeNet5	3	0.998	0.997	0.999	7.21	5.88	9.02
	LeNet5-(fc1-80)	3	0.998	0.996	0.999	7.80	6.77	11.01
	LeNet5-(fc1-100)	3	0.997	0.995	0.999	7.42	6.20	9.10
	LeNet5-(fc1-150)	3	0.998	0.992	0.999	7.35	5.34	9.62
	VGG11-W32	1	0.990	0.988	0.993	6.02	5.57	6.51
	VGG11-W64	1	0.996	0.993	0.999	5.87	5.32	6.26
	VGG11-W64	1	0.995	0.993	0.996	6.24	5.97	6.70
CIFAR100	VGG11-W48	1	0.999	0.999	0.999	17.861	15.456	20.491
	VGG11-W64	1	0.999	0.999	1.000	19.185	18.358	20.410
	VGG11-W80	1	0.999	0.999	1.000	19.455	18.120	21.450
	ResNet18-W48	1	1.000	1.000	1.000	28.23	27.37	29.27
	ResNet18-W64	1	1.000	1.000	1.000	27.07	25.72	29.50
	ResNet18-W80	1	1.000	1.000	1.000	28.23	25.98	30.03

Table C.6: Squared dot product $(\boldsymbol{v}_1^{\mathsf{T}}\widehat{\mathbb{E}}[\boldsymbol{x}])^2$ and spectral ratio λ_1/λ_2 for convolutional layers in the selection of network structures and datasets in Table C.5.

			$(\boldsymbol{v}_1^{\mathsf{T}}\widehat{\mathbb{E}}[\boldsymbol{x}])^2$			λ_1/λ_2		
Dataset	Network	# conv	mean	min	max	mean	min	max
CIFAR10	LeNet5	1	0.999	0.998	0.999	15.87	11.15	27.20
	LeNet5-(fc1-80)	1	0.998	0.998	0.999	12.36	9.53	13.36
	LeNet5-(fc1-100)	1	0.999	0.999	0.999	19.49	16.69	21.92
	LeNet5-(fc1-150)	1	0.999	0.998	0.999	12.86	7.65	16.34
	VGG11-W32	7	0.995	0.991	0.999	5.31	2.39	9.09
	VGG11-W64	7	0.997	0.993	1.000	5.76	2.50	9.98
	VGG11-W64	7	0.998	0.995	1.000	5.81	2.53	10.62
CIFAR100	VGG11-W48	7	0.996	0.991	0.999	5.72	2.46	9.90
	VGG11-W64	7	0.995	0.991	0.999	5.66	2.50	10.79
	VGG11-W80	7	0.994	0.988	0.998	5.18	2.50	8.45
	ResNet18-W48	19	0.981	0.917	0.998	3.79	1.89	7.56
	ResNet18-W64	19	0.985	0.910	0.998	3.96	1.81	7.53
	ResNet18-W80	19	0.987	0.954	0.997	4.16	2.11	7.04

C.6.2 Eigenspace Overlap Between Different Models

The non trivial overlap between top eigenspaces of layer-wise Hessians is one of our interesting observations that had been discusses in Section 4.5.3. Here we provide more related empirical results. Some will further verify our claim in Section 4.5.3 and some will appear to be challenge that. Both results will be explained discussed more extensively in Section C.7.

Overlap preserved when varying hyper-parameters:

We first verify that the overlap also exists for a set of models trained with the different hyper-parameters. Using the LeNet5 (defined in Table C.3) as the network structure. We train 6 models using the default training scheme (SGD, lr=0.01, momentum=0), 5 models using a smaller learning rate (SGD, lr=0.001, momentum=0), and 5 models using a combination of optimization tricks (SGD, lr=0.01, momentum=0.9, weight decay=0.0005). With these 16 models, we compute the pairwise eigenspace overlap of their layer-wise Hessians (120 pairs in total) and plot their average in Figure C.2. The shade areas in the figure represents the standard deviation. The pattern of overlap is clearly preserved, and the position of the peak roughly agrees with the output dimension m, demonstrating that the phenomenon is caused by a common structure instead of similarities in training process.

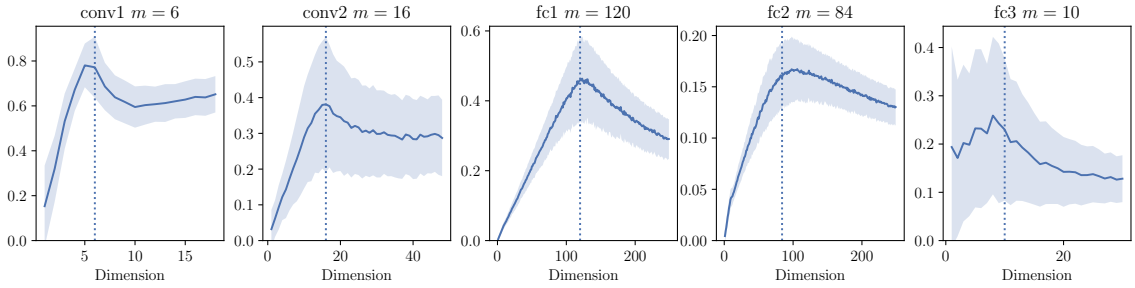


FIGURE C.2: Eigenspace overlap of different models of LeNet5 trained with different hyperparameters.

Note that for fc3 (the final output layer), we are not observing a linear growth starting from 0 like other layers. This can be explained by the lack of neuron permutation. Related details will be discussed along with the reason for the linear growth pattern for other layers in Section C.7.2.

Eigenspace overlap for convolutional layers in large models:

Even though the exact Kroneckor Factorization for layer-wise Hessians is only well-defined for fully connected layers, we also observe similar nontrivial eigenspace overlap for convolutional layers in larger and deeper networks including variants of VGG11 and ResNet18 on datasets CIFAR10 and CIFAR100. Some representative results are shown in Figure C.3 and Figure C.4. For each model on each dataset, we independently train 5 models and compute the average pairwise eigenspace overlap. The shade areas represents the standard deviation.

For most of the convolutional layers, the eigenspace overlap peaks around the dimension which is equal to the number of output channels of that layer, which is similar to the layers in LeNet5 as in Figure C.2. The eigenspace overlap of the final fully connected-layer also behaves similar to fc3:LeNet5, which remains around a constant then drops after exceeding the dimension of final output. However, there are also layers whose overlap does not peak around the output dimensions, (e.g. conv2 of Figure C.3(a) and conv7 of Figure C.4(a)). We will discuss these special cases in the following paragraph.

Failed cases for eigenspace overlap

As seen in Figure C.3 and Figure C.4, there is a small portion of layers, usually closer to the input, whose eigenspace overlap does peak around the output dimensions. These layers can be clustered into the following two general cases.

Early Peak of Low Overlap For layers shown in Figure C.5. The overlap of dominating eigenspaces are significantly lower than the other layers. Also there exists a small peak at very small dimensions.

Delayed Peak / Peak Doesn't Decline For layers shown in Figure C.6, the top eigenspaces has a nontrivial overlap, but the peak dimension is larger than predicted output dimension.

However, the existence of such failure cases *does not* undermine the theory of Kronecker factorization approximation. In fact, both appear because the top hessian eigenspace is not completely spanned by $\mathbb{E}[\boldsymbol{x}]$, and can be predicted by computing the auto correlation matrices and the output Hessians. The details will also be elaborated in Section C.7.2 with the help of correspondence matrices.

C.6.3 Eigenvector Correspondence

In this section, we leverage the idea of eigenvector matricization (Definition 4.2.2) and analyze the validity of the decoupling conjecture using a matrix which we defined as the eigenvector corresponding matrix. First let us recall the definition of eigenvector matricization

Definition 4.2.2 *Consider a layer with input dimension n and output dimension m. For an eigenvector $\boldsymbol{h} \in \mathbb{R}^{mn}$ of its layer-wise Hessian, the matricized form of $\boldsymbol{h}$ is* $\mathrm{Mat}(\boldsymbol{h}) \in \mathbb{R}^{m\times n}$ *where* $\mathrm{Mat}(\boldsymbol{h})_{i,j} = \boldsymbol{h}_{(i-1)m+j}$.

Suppose the i-th eigenvector for $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ is $\boldsymbol{v}_i$ and the j-th eigenvector for $\mathbb{E}[\boldsymbol{M}]$ is $\boldsymbol{u}_j$. Then the Kronecker product $\mathbb{E}[\boldsymbol{M}]\otimes\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ has an eigenvector $\boldsymbol{u}_j\otimes\boldsymbol{v}_i$. Therefore if the decoupling conjecture is true, one would expect that the top eigenvector of the layer-wise Hessian have a clear correspondence with the top eigenvectors of its

two components. Note that $\boldsymbol{u} \otimes \boldsymbol{v}$ is just the flattened matrix $\boldsymbol{u}\boldsymbol{v}^\mathsf{T}$.

More concretely, to demonstrate the correspondence between the eigenvectors of the layerwise hessian and the eigenvectors of matrix $\mathbb{E}[\boldsymbol{M}]$ and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$, we introduce "eigenvector correspondence matrices" as shown in Figure C.7.

Definition C.6.1 (Eigenvector Correspondence Matrices)**.** For layer-wise Hessian matrix $\boldsymbol{H} \in \mathbb{R}^{mn \times mn}$ with eigenvectors $\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{mn}$, and its corresponding auto-correlation matrix $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}] \in \mathbb{R}^{n \times n}$ with eigenvectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n$. The correspondence between $\boldsymbol{v}_i$ and $\boldsymbol{h}_j$ can be defined as

$$\mathrm{Corr}(\boldsymbol{v}_i, \boldsymbol{h}_j) := \| \mathrm{Mat}(\boldsymbol{h}_j)\boldsymbol{v}_i\|^2. \tag{C.258}$$

For the output Hessian matrix $\mathbb{E}[\boldsymbol{M}] \in \mathbb{R}^{m \times m}$ with eigenvectors $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m$, we can likewise define correspondence between $\boldsymbol{v}_i$ and $\boldsymbol{h}_j$ as

$$\mathrm{Corr}(\boldsymbol{u}_i, \boldsymbol{h}_j) := \| \mathrm{Mat}(\boldsymbol{h}_j)^\mathsf{T}\boldsymbol{u}_i\|^2 \tag{C.259}$$

We may then define the eigenvector correspondence matrix between $\boldsymbol{H}$ and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ as a $n \times mn$ matrix whose i, j-th entry is $\mathrm{Corr}(\boldsymbol{v}_i, \boldsymbol{h}_j)$, and the eigenvector correspondence matrix between $\boldsymbol{H}$ and $\mathbb{E}[\boldsymbol{M}]$ as a $m \times mn$ matrix whose i, j-th entry is $\mathrm{Corr}(\boldsymbol{u}_i, \boldsymbol{h}_j)$.

Intuitively, if the i, j-th entry of the corresponding matrix is close to 1, then the eigenvector $\boldsymbol{h}_j$ is likely to be the Kronecker product of $\boldsymbol{v}_i$ (or $\boldsymbol{u}_i$) with some vector. Note that if the decoupling conjecture holds absolutely, every eigenvector of the layer-wise Hessian (column of the correspondence matrices) should have a perfect correlation of 1 with exactly one of $\boldsymbol{v}_i$ and one of $\boldsymbol{u}_i$. In Figure C.7 we can see that the correspondence matrices for the true layer-wise Hessian approximately satisfies this property for top eigenvectors. The similarity between the correspondence patterns

for the true and approximated Hessian also verifies the validity of the Kronecker approximation for dominating eigenspace.

In Figure C.7, we show the heatmap of Eigenvector Correspondence Matrices for fc1:LeNet5, which has 120 output neurons. Here we take the top left corner of the eigenvector correspondence matrices. We can see that the top 120 eigenvectors of $\mathbb{E}[\boldsymbol{H}]$, roughly corresponds to the top 120 eigenvectors of $\mathbb{E}[\boldsymbol{M}]$ (as shown by the diagonal patter of (b)) and the first eigenvector of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ (as shown by the horizontal pattern of (a)). The similarity between the first row and the second row also shows the validity of the Kronecker approximation.

Here we present the correspondence matrix for fc2, conv1, and conv2 layer of LeNet5. The top eigenvectors for all layers shows a strong correlation with the first eigenvector of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ (which is approximately $\widehat{\mathbb{E}}[\boldsymbol{x}]$). For convolutional layers, since the computation of $\boldsymbol{M}$ is not exact, the correspondence matrices with $\mathbb{E}[\boldsymbol{M}]$ does not exhibit the diagonal pattern. For fc2:LeNet5 as in Figure C.9, the diagonal pattern in (b) and the strong correlation with $\mathbb{E}[\boldsymbol{x}]$ stops at dimension 9. This fells into one of the "failed cases" as described in Section C.6.2 case that the small eigenvalues of $\mathbb{E}[\mathbf{M}]$ are approaching 0 faster than $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$. We will discuss this case in more detail in Section C.7.2.

For VGG11 we also observe a strong correlation with the first eigenvector of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$.

C.6.4 Structure of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ and $\mathbb{E}[\boldsymbol{M}]$ During Training

We observed the pattern of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ matrix and $\mathbb{E}[\boldsymbol{M}]$ matrix along the training trajectory (Figure C.15, Figure C.16). It shows that $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ is always approximately rank 1, and $\mathbb{E}[\boldsymbol{M}]$ always have around c large eigenvalues. According to our analysis,

since the nontrivial eigenspace overlap is likely to be a consequence of a approximately rank 1 $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$, we would conjecture that the overlap phenomenon is likely to happen on the training trajectory as well.

C.7 Additional Explanations

C.7.1 Dominating Eigenvectors of Layer-Wise Hessian are Low Rank

A natural corollary for the Kronecker factorization approximation of layer-wise Hessians is that the eigenvectors of the layer-wise Hessians are low rank. Let $\boldsymbol{h}_i$ be the i-th eigenvector of a layer-wise Hessian. The rank of $\mathrm{Mat}(\boldsymbol{h}_i)$ can be considered as an indicator of the complexity of the eigenvector. Consider the case that $\boldsymbol{h}_i$ is one of the top eigenvectors. From Section 4.5.3, we have $\boldsymbol{h}_i \approx \boldsymbol{u}_i \otimes \widehat{\mathbb{E}}[\boldsymbol{x}]$. Thus, $\mathrm{Mat}(\boldsymbol{h}_i) \approx \boldsymbol{u}_i \widehat{\mathbb{E}}[\boldsymbol{x}]^\mathsf{T}$, which is approximately rank 1. Experiments shows that first singular values of $\mathrm{Mat}(\boldsymbol{h}_i)$ divided by its Frobenius Norm are usually much larger than 0.5, indicating the top eigenvectors of the layer-wise Hessians are very close to rank 1. Figure C.17 shows first singular values of $\mathrm{Mat}(\boldsymbol{h}_i)$ divided by its Frobenius Norm for i from 1 to 200. We can see that the top eigenvectors of the layer-wise Hessians are very close to rank 1.

C.7.2 Eigenspace Overlap of Different Models

From the experiment results in Section C.6 together with Figure 4.4, we can see that our approximation and explanation stated in Section 4.5.3 of the main text is approximately correct but may not be so accurate for some layers. We now present a more general explanation which addresses why the overlap before rank-m grows linearly. We will also explain some exceptional cases as shown in Section C.6.2 and possible discrepancies of our approximation.

Let $\boldsymbol{h}_i$ be the i-th eigenvector of the layer-wise Hessian $\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)})$, under the assumption that the autocorrelation matrix $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ is approximately rank 1 that $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}] \approx \mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{x}]^\mathsf{T}$, for all $i \leqslant m$, we can approximate the $\boldsymbol{h}_i$ as $\boldsymbol{u}_i \otimes (\mathbb{E}[\boldsymbol{x}]/\|\mathbb{E}[\boldsymbol{x}]\|)$ where $\boldsymbol{u}_i$ is the i-th eigenvector of $\mathbb{E}[\boldsymbol{M}]$. Formally, the trend of top eigenspace can

be characterized by the following theorem. For simplicity of notations, we abuse the superscript within parentheses to refer the two models instead of layer number in this section.

Theorem C.7.1. *Consider 2 different models with the same network structure trained on the same dataset. Fix the p-th hidden layer with input dimension n and output dimension m. For the first model, denote its output Hessian as $\mathbb{E}[\boldsymbol{M}]^{(1)}$ with eigenvalues $\tau_1^{(1)} \geqslant \tau_2^{(1)} \geqslant \cdots \geqslant \tau_m^{(1)} \geqslant 0$ and eigenvectors $\boldsymbol{r}_1^{(1)}, \cdots, \boldsymbol{r}_m^{(1)} \in \mathbb{R}^m$; denote its autocorrelation matrix as $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]^{(1)}$, with eigenvalues $\gamma_1^{(1)} \geqslant \gamma_2^{(1)} \geqslant \cdots \geqslant \gamma_m^{(1)} \geqslant 0$ and eigenvectors $\boldsymbol{t}_1^{(1)}, \cdots, \boldsymbol{t}_n^{(1)} \in \mathbb{R}^n$. The variables for the second matrices are defined identically by changing 1 in the superscript parenthesis to 2.*

Assume the Kronecker factorization approximation is accurate that $\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)})^{(1)} \approx \mathbb{E}[\boldsymbol{M}]^{(1)} \otimes \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]^{(1)}$ and $\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)})^{(2)} \approx \mathbb{E}[\boldsymbol{M}]^{(2)} \otimes \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]^{(2)}$. Also assume the autocorrelation matrices of two models are sufficiently close to rank 1 in the sense that $\tau_m^{(1)}\gamma_1^{(1)} > \tau_1^{(1)}\gamma_2^{(1)}$ and $\tau_m^{(2)}\gamma_1^{(2)} > \tau_1^{(2)}\gamma_2^{(2)}$. Then for all $k \leqslant m$, the overlap of top k eigenspace between their layerwise Hessians $\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)})^{(1)}$ and $\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)})^{(2)}$ will be approximately $\frac{k}{m}(\boldsymbol{t}_1^{(1)} \cdot \boldsymbol{t}_1^{(2)})^2$. Consequently, the top eigenspace overlap will show a linear growth before it reaches dimension m. The peak at m is approximately $(\boldsymbol{t}_1 \cdot \boldsymbol{t}_2)^2$.

Proof of Theorem C.7.1. Let $\boldsymbol{h}_i^{(2)}$ be the i-th eigenvector of the layer-wise Hessian for the first model $\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)})^{(1)}$, and $\boldsymbol{g}_i$ be that of the second model $\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)})^{(2)}$. Consider the first model. By the Kronecker factorization approximation, since $\tau_m^{(1)}\gamma_1^{(1)} > \tau_1^{(1)}\gamma_2^{(1)}$, the top m eigenvalues of the layer-wise Hessian are $\gamma_1^{(1)}\tau_1^{(1)}, \cdots, \gamma_1^{(1)}\tau_m^{(1)}$. Consequently, for all $i \leqslant m$ we have $\boldsymbol{h}_i \approx \boldsymbol{r}_i^{(1)\mathsf{T}} \otimes \boldsymbol{t}_1^{(1)}$. Thus, for any $k \leqslant m$, we have its top k eigenspace as $\boldsymbol{V}_k^{(1)} \otimes \boldsymbol{t}_1^{(1)}$, where $\boldsymbol{V}_k^{(1)} \in \mathbb{R}^{m \times k}$ has column vectors $\boldsymbol{r}_1^{(1)}, \ldots, \boldsymbol{r}_k^{(1)}$. Similarly, for the second model we have $\boldsymbol{h}_i^{(2)} \approx \boldsymbol{r}_i^{(2)} \otimes \boldsymbol{t}_1^{(2)}$ and the

top k eigenspace as $\boldsymbol{V}_k^{(2)} \otimes \boldsymbol{t}_1^{(2)}$, where $\boldsymbol{V}_k^{(2)}$ has column vectors $\boldsymbol{r}_1^{(2)}, \ldots, \boldsymbol{r}_k^{(2)}$. The eigenspace overlap of the 2 models at dimension k is thus

$$\begin{aligned} \text{Overlap}\left(\boldsymbol{V}_k^{(1)} \otimes \boldsymbol{t}_1^{(1)}, \boldsymbol{V}_k^{(2)} \otimes \boldsymbol{t}_1^{(2)}\right) &= \frac{1}{k}\left\|\boldsymbol{V}_k^{(1)\mathsf{T}}\boldsymbol{V}_k^{(2)} \otimes \boldsymbol{t}_1^{(1)\mathsf{T}}\boldsymbol{t}_1^{(2)}\right\|_F^2 \\ &= \left(\boldsymbol{t}_1^{(1)} \cdot \boldsymbol{t}_1^{(2)}\right)^2 \text{Overlap}\left(\boldsymbol{V}_k^{(1)}, \boldsymbol{V}_k^{(2)}\right). \end{aligned} \tag{C.260}$$

Note that for all $i \leqslant m$, $\boldsymbol{r}_i^{(1)}, \boldsymbol{r}_i^{(2)} \in \mathbb{R}^n$, which is the space corresponding to the neurons. Since for hidden layers, the output neurons (channels for convolutional layers) can be arbitrarily permuted to give equivalent models while changing eigenvectors. For $\boldsymbol{h}_i \approx \boldsymbol{r}_i \otimes \boldsymbol{t}_1$, permuting neurons will permute entries in $\boldsymbol{r}_i$. Thus, we can assume that for two models, $\boldsymbol{r}_i^{(1)}$ and $\boldsymbol{r}_i^{(2)}$ are not correlated and thus have an expected inner product of $\sqrt{1/m}$.

It follows from Definition 4.2.1 that

$$\mathbb{E}[\text{Overlap}(\boldsymbol{V}_k^{(1)}, \boldsymbol{V}_k^{(2)})] = \sum_{i=1}^{k} \mathbb{E}[(\boldsymbol{r}_i^{(1)} \cdot \boldsymbol{r}_i^{(2)})^2] = k(\frac{1}{m}) = \frac{k}{m} \tag{C.261}$$

and thus the eigenspace overlap of at dimension k would be approximately $\frac{k}{m}(\boldsymbol{t}_1^{(1)} \cdot \boldsymbol{t}_1^{(2)})^2$. This explains the peak at dimension m and the linear growth before it. □

From our results on autocorrelation matrices in Section 4.3.1 and Section C.6.1, we have $\widehat{\mathbb{E}}[\boldsymbol{x}]^{(1)} \approx \boldsymbol{t}_1^{(1)}$ and $\widehat{\mathbb{E}}[\boldsymbol{x}]^{(2)} \approx \boldsymbol{t}_1^{(2)}$ where $\widehat{\mathbb{E}}$ is the normalized expectation. Hence when $k = m$, the overlap is approximately $(\widehat{\mathbb{E}}[\boldsymbol{x}]^{(1)} \cdot \widehat{\mathbb{E}}[\boldsymbol{x}]^{(2)})^2$. Since $\widehat{\mathbb{E}}[\boldsymbol{x}]^{(1)}$ and $\widehat{\mathbb{E}}[\boldsymbol{x}]^{(2)}$ are the identical for the input layers, the overlap is expected to be very high at dimension m for input layers. For other hidden layers in a ReLU network, $\boldsymbol{x}$ are output of ReLU and thus non-negative. Two non-negative vectors $\widehat{\mathbb{E}}[\boldsymbol{x}]^{(1)}$ and $\widehat{\mathbb{E}}[\boldsymbol{x}]^{(2)}$ still have relatively large dot product, which contributes to the high overlap

peak.

The Decreasing Overlap After Output Dimension

Consider the $(m+1)$-th eigenvector $\boldsymbol{h}_{m+1}^{(1)}$ of the first model. Following the Kronecker factorization approximation and assumptions in Theorem C.7.1, we have $\boldsymbol{h}_{m+1}^{(1)} \approx \boldsymbol{r}_1^{(1)} \otimes \boldsymbol{t}_2^{(1)}$. Since top m eigenspace of the first model is approximately $\boldsymbol{I}_m \otimes \boldsymbol{t}_1^{(1)}$ and $\boldsymbol{t}_2^{(1)}$ is orthogonal to $\boldsymbol{t}_1^{(1)}$, the $\boldsymbol{h}_{m+1}^{(1)}$ eigenvector will be orthogonal to the top m eigenspace of the first model. It will also have low overlap with $\boldsymbol{I}_m \otimes \boldsymbol{t}_1^{(2)}$ since $(\widehat{\mathbb{E}}[\boldsymbol{x}]^{(1)} \cdot \widehat{\mathbb{E}}[\boldsymbol{x}]^{(2)})^2$ is large.

Moreover, since the remaining eigenvectors of the autocorrelation matrix no longer has the all positive property as the first eigenvector and structure of the convariance $\Sigma_{\boldsymbol{x}}$ is directly associated with the ordering of the input neurons which are randomly permuted across different models, the overlap between other eigenvectors of the autocorrelation matrix across different models will be close to random, hence the overlap after the top m dimension will decrease until the eigenspaces has sufficiently many basis vectors to make the random overlap large.

The Output Layer

Note that for the last layer satisfying the assumptions in Theorem C.7.1, the overlap will stay high before dimension m and be approximately $(\boldsymbol{t}_1 \cdot \boldsymbol{t}_2)^2$ since the output neurons directly correspondence to classes, and hence neurons cannot be permuted. In this case, the overlap will be approximately $(\boldsymbol{t}_1 \cdot \boldsymbol{t}_2)^2$ for all dimension $k \leqslant m$. This is consistent with our observations.

Explaining "Failed Cases" of Eigenspace Overlap

As shown in Figure C.5 and Figure C.6, the nontrivial top eigenspace overlap does not necessarily peak at the output dimension for all layers. Some layers has a low peak at very small dimensions and others has a peak at a larger dimension. With the more complete analysis provided above, we now proceed to explain these two phenomenons. The major reason for such phenomenons is that the assumption of autocorrelation matrix being sufficiently close to rank 1 is not always satisfied. In particular, following the notations in Theorem C.7.1, for these exceptional layers we have $\tau_m\gamma_1 < \tau_1\gamma_2$. We first consider the first phenomenon (early peak of low overlap) and take fc2:F-200^2 (MNIST) in as an example. Here Figure C.19(a) is identical to Figure C.5(a), which displays the early peak around $m = 10$.

As shown in Figure C.19(b), the second eigenvalue of the auto correlation $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ is as large as approximately 1/10 of the first eigenvalue. With the output Hessian have $c - 1 = 9$ significant large eigenvalues as described in Section 4.5.2, it has $\tau_{10}\gamma_1 < \tau_1\gamma_2$. Thus through the Kronecker factorization approximation, the top m dimensional eigenspace is no longer simply $\boldsymbol{I}_m \otimes \widehat{\mathbb{E}}[\boldsymbol{x}]$, but a subset of top eigenvectors of the output Hessian Kroneckered with a subset of top eigenvectors of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ as reflected in Figure C.19(d). This "mixture" of Kronecker product is moreover verified in Figure C.19(c).

As reflected by the first row of Figure C.19(c) and Figure C.19(d), for $i \leqslant 9$ we have $\boldsymbol{h}_i \approx \boldsymbol{r}_i \otimes \widehat{\mathbb{E}}[\boldsymbol{x}]$, which falls in the regime of Theorem C.7.1. Hence we are seeing an linearly growing pattern of the overlap for dimension less than 10 and reaches a mean overlap of around 0.012 by dimension 9. If following this linear trend, the overlap would be close to 0.25 by the output dimension of 200. However, since the 10-th eigenvalue of the output Hessian is significantly smaller, little of

the 10-19 dimensional eigenspace were contributed by $\widehat{\mathbb{E}}[\boldsymbol{x}]$, hence the overlap of dimension larger than 10 falls into the regime discussed in Section C.7.2, for which we see a sharp decrease of overlap after dimension 9. Note that this example shows that Kronecker factorization can be used to predict when our conditions in Theorem C.7.1 fails and also predict the condition can be satisfied up to which dimension. As shown in Figure C.20, similar explanation also applies to convolutional layers in larger networks.

We then consider the second phenomenon (delayed peak) and take conv2:VGG11-W200 (CIFAR10) in as an example. Here Figure C.21(a) is identical to Figure C.6(d), which has the overlap peak later than the output dimension 200. In this case, the second eigenvalue of the auto correlation matrix is still not negligible compared to the top eigenvalue. What differentiate this case from the first phenomenon is that the eigenvalues of the output Hessian no longer has a significant peak – instead it has a heavy tail which is necessary for high overlap.

Towards dimension m there gradually exhibits higher correspondence to later eigenvectors of the input autocorrelation matrix and hence less correspondence to $\widehat{\mathbb{E}}[\boldsymbol{x}]$. This eventually results in the delayed and flattened peak.

Since the full correspondence matrices are too large to be visualized, we plotted their first rows up to 400 dimensions in Figure C.21(e) and Figure C.21(f), in which each dot represents the average of correlation with $\widehat{\mathbb{E}}[\boldsymbol{x}]$ for the 10 eigenvector nearby. From these figures it is straightforward to see the gradual decreasing correlation with $\widehat{\mathbb{E}}[\boldsymbol{x}]$.

C.7.3 Batch Normalization and Zero-Mean Input

In this section, we show the results on networks with using Batch normalization (BN) (Ioffe and Szegedy, 2015). For layers after BN, we have $\mathbb{E}[\boldsymbol{x}] \approx 0$ so that $\mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{x}]^\mathsf{T}$ no longer dominates $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ and the low rank structure of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ should disappear. Thus, we can further expect that the overlap between top eigenspace of layer-wise Hessian among different models will not have a peak.

Table C.7 shows the same experiments done in Table C.5. The values for each network are the average of 3 different models. It is clear that the high inner product and large spectral ratio both do not hold here, except for the first layer where there is no normalization applied. Note that we had channel-wise normalization (zero-mean for each channel but not zero-mean for $\boldsymbol{x}$) for conv1 in LeNet5 so that the spectral ratio is also small.

Table C.7: Structure of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ for BN networks

			$(v_1^\mathsf{T}\widehat{\mathbb{E}}[\boldsymbol{x}])^2$			λ_1/λ_2		
Dataset	Network	# fc	mean	min	max	mean	min	max
MNIST	F-200^2-BN	2	0.062	0.001	0.260	1.16	1.04	1.30
	F-600^2-BN	2	0.026	0.000	0.063	1.13	1.02	1.26
	F-600^4-BN	4	0.027	0.000	0.146	1.11	1.03	1.19
CIFAR10	LeNet5-BN	3	0.210	0.001	0.803	1.54	1.20	1.89

Figure C.23(a) shows that $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ is no longer close to rank 1 when having BN. This is as expected. However, $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ still has a few large eigenvalues.

Figure C.23(b) shows the eigenvector correspondence matrix of True Hessian with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ for fc1:LeNet5. Because $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ is no longer close to rank 1, only very few eigenvectors of the layer-wise Hessian will have high correspondence with the top eigenvector of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$, as expected. This directly leads to the disappearance of peak

in top eigenspace overlap of different models, as shown in Figure C.23. The peak still exists in conv1 because BN is not applied to the input.

Comparing Figure C.23(b) and Figure C.23(c), we can see that the Kronecker factorization still gives a reasonable approximation for the eigenvector correspondence matrix with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$, although worse than the cases without BN (Figure C.7).

Figure C.24 compare the eigenvalues and top eigenspaces of the approximated Hessian and the true Hessian for LeNet5 with BN. The approximation using Kronecker factorization is also worse than the case without BN (Figure 4.2). However, the approximation still gives meaningful information as the overlap of top eigenspace is still highly nontrivial.

C.7.4 Outliers in Hessian Eigenspectrum

One characteristic of Hessian that has been mentioned by many is the outliers in the spectrum of eigenvalues. Sagun et al. (2018) suggests that there is a gap in Hessian eigenvalue distribution around the number of classes c in most cases, where $c = 10$ in our case. A popular theory to explain the gap is the class / logit clustering of the logit gradients (Fort and Ganguli, 2019; Papyan, 2019, 2020). Note that these explanations can be consistent with our heuristic formula for the top eigenspace of output Hessian at initialization– in the two-layer setting we considered the logit gradients are indeed clustered.

In the layer-wise setting, the clustering claim can be formalized as follows: For each class $k \in [c]$ and logit entry $l \in [c]$, with $\boldsymbol{Q}$ be defined as in Equation C.17, and $(\mathbf{x}, \mathrm{y})$ as the input, label pair, let

$$\boldsymbol{\Delta}_{i,j} = \mathbb{E}\left[\boldsymbol{Q}_{\mathbf{x}} \frac{\partial \boldsymbol{z}_{\mathbf{x}}}{\partial \boldsymbol{w}_j^{(p)}} \middle| \mathrm{y} = i \right]. \tag{C.262}$$

Then at the initialization, for each logit entry j, $\{\boldsymbol{\Delta}_{i,j}\}_{i\in[c]}$ is clustered around the "logit center" $\widehat{\boldsymbol{\Delta}}_j \triangleq \mathbb{E}_{i\in[c]}[\boldsymbol{\Delta}_{i,j}]$; at the minima, for each class i, $\{\boldsymbol{\Delta}_{i,j}\}_{j\in[c]}$ is clustered around the "class center" $\widehat{\boldsymbol{\Delta}}_i \triangleq \mathbb{E}_{j\in[c]}[\boldsymbol{\Delta}_{i,j}]$. With the decoupling conjectures, we may also consider similar claims for output Hessians, where

$$\boldsymbol{\Gamma}_{i,j} = \mathbb{E}\left[\boldsymbol{Q}_{\mathbf{x}} \frac{\partial \boldsymbol{z}_{\mathbf{x}}}{\partial \boldsymbol{z}_{\mathbf{x}}^{(p)}{}_j} \middle| \mathrm{y} = i \right]. \tag{C.263}$$

A natural extension of the clustering phenomenon on output Hessians is then as follows: At the initialization, for each logit entry j, $\{\boldsymbol{\Gamma}_{i,j}\}_{i\in[c]}$ is clustered around $\widehat{\boldsymbol{\Gamma}}_j \triangleq \mathbb{E}_{i\in[c]}[\boldsymbol{\Gamma}_{i,j}]$; at the minima, for each class i, $\{\boldsymbol{\Gamma}_{i,j}\}_{j\in[c]}$ is clustered around $\widehat{\boldsymbol{\Gamma}}_i \triangleq$

$\mathbb{E}_{j\in[c]}[\boldsymbol{\Gamma}_{i,j}]$. Note that we have the layer-wise Hessian and layer-wise output Hessian satisfying

$$\boldsymbol{H}_{\mathcal{L}}(\boldsymbol{w}^{(p)}) = \mathop{\mathbb{E}}_{i,j\in[c]}[\boldsymbol{\Delta}_{i,j}^{\mathsf{T}}\boldsymbol{\Delta}_{i,j}], \quad \boldsymbol{M}^{(p)} = \mathop{\mathbb{E}}_{i,j\in[c]}[\boldsymbol{\Gamma}_{i,j}^{\mathsf{T}}\boldsymbol{\Gamma}_{i,j}]. \tag{C.264}$$

Low-rank Hessian at Random Initialization and Logit Gradient Clustering

We first briefly recapture our explanation on the low-rankness of Hessian at random initialization. In Section 4.4 and Section C.2.2, we have shown that for a two layer ReLU network with Gaussian random initialization and Gaussian random input, the output hessian of the first layer $\boldsymbol{M}^{(1)}$ is approximately $\frac{1}{4}\boldsymbol{W}^{(2)T}\boldsymbol{A}\boldsymbol{W}^{(2)}$. We then heuristically extend this approximation to a randomly initialized L-layer network, that with $\boldsymbol{S}^{(p)} = \boldsymbol{W}^{(L)}\boldsymbol{W}^{(L-1)}\cdots\boldsymbol{W}^{(p+1)}$, the output Hessian of the p-th layer $\boldsymbol{H}^{(p)}$ can be approximated by $\widetilde{\boldsymbol{M}}^{(p)}$ where

$$\widetilde{\boldsymbol{M}}^{(p)} \triangleq \frac{1}{4^{L-p}}\boldsymbol{S}^{(p)T}\boldsymbol{A}\boldsymbol{S}^{(P)}. \tag{C.265}$$

Since $\boldsymbol{A}$ is strictly rank $c-1$ with null space of the all-one vector, $\boldsymbol{H}^{(p)}$ is strictly rank $c-1$. Thus $\boldsymbol{H}^{(p)}$ is approximately rank $c-1$, and so is the corresponding layerwise Hessian according to the decoupling conjecture.

Now we discuss the connection between our analysis with the theory of logit gradient clustering. As previously observed by Papyan (2019), for each logit entry l, $\{\boldsymbol{\Delta}_{i,j}\}_{l\in[c]}$ are clustered around the logit gradients $\mathbb{E}_{l\in[c]}[\boldsymbol{\Delta}_{i,j}]$. Similar clustering effects for $\{\boldsymbol{\Gamma}_{i,j}\}_{l\in[c]}$ were also empirically observed by our experiments. Moreover, through the approximation above and the decoupling conjecture, for each logit entry j, the cluster centers $\widehat{\boldsymbol{\Gamma}}_j$ and $\widehat{\boldsymbol{\Delta}}_j$ can be approximated by

$$\begin{aligned}\widehat{\boldsymbol{\Gamma}}_j &\approx \breve{\boldsymbol{\Gamma}}_j \triangleq (\boldsymbol{S}^{\mathsf{T}}\boldsymbol{Q})_j \\ \widehat{\boldsymbol{\Delta}}_j &\approx \breve{\boldsymbol{\Delta}}_j \triangleq ((\mathbb{E}[\boldsymbol{x}]\otimes\boldsymbol{S}^{\mathsf{T}})\mathbb{E}[\boldsymbol{Q}])_j.\end{aligned} \tag{C.266}$$

Following Papyan (2019), we used t-SNE (Van der Maaten and Hinton, 2008) to visualize the logit gradients. As we see in Figure C.25, the "logit centers" of the clustering directly corresponds to the approximated dominating eigenvectors of the Hessian, which is consistent with our analysis.

Gradient Clustering at Minima Currently our theory does not provide an explanation to the low rank structure of Hessian at the minima. However we have observed that the class clustering of logit gradients does not universally apply to all models at the minima, even when the models have around c significant large eigenvalues. As shown in Figure C.26, the class clustering is very weak but there are still around c significant large eigenvalues. We conjecture that the class clustering of logit gradients may be a sufficient but not necessary condition for the Hessian to be low rank at minima.

C.8 Computing PAC-Bayes Bounds with Hessian Approximation

Given a model parameterized with θ and an input-label pair $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d \times \mathbb{R}^c$, the classification error of θ over the input sample $\boldsymbol{x}$ is $\breve{l}(\theta, \boldsymbol{x}) := \mathbf{1}[\arg\max f_\theta(\boldsymbol{x}) = \arg\max \boldsymbol{y}]$. With the underlying data distribution D and training set S i.i.d. sampled from D, we define

$$e(\theta) := \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D}[\breve{l}(\theta, \boldsymbol{x})], \qquad \widehat{e}(\theta) := \frac{1}{N}\sum_{i=1}^{N}[\breve{l}(\theta, \boldsymbol{x}_i)] \tag{C.267}$$

as the expected and empirical classification error of θ, respectively. We define the measurable hypothesis space of parameters $\mathcal{H} := \mathbb{R}^P$. For any probabilistic measure P in $\mathcal{H}$, let $e(P) = \mathbb{E}_{\theta\sim P} e(\theta)$, $\widehat{e}(P) = \mathbb{E}_{\theta\sim P}\widehat{e}(\theta)$, and $\breve{e}(P) = \mathbb{E}_{\theta\sim P}\mathcal{L}(\theta)$. Here $\breve{e}(P)$ serves as a differentiable convex surrogate of $\widehat{e}(P)$.

Theorem C.8.1 (Pac-Bayes Bound)**.** *(McAllester, 1999)(Langford and Seeger, 2001) For any prior distribution P in $\mathcal{H}$ that is chosen independently from the training set S, and any posterior distribution Q in $\mathcal{H}$ whose choice may inference S, with probability $1-\delta$,*

$$\mathrm{KL}\left(\widehat{e}(Q) \| e(Q)\right) \leqslant \frac{\mathrm{KL}(Q\|P) + \log\frac{|S|}{\delta}}{|S|-1}. \tag{C.268}$$

Fix some constant $b, c \geqslant 0$ and $\theta_0 \in \mathcal{H}$ as a random initialization, Dziugaite and Roy (2017) shows that when setting $Q = \mathcal{N}(\boldsymbol{w}, \mathrm{diag}(\boldsymbol{s}))$, $P = \mathcal{N}(\theta_0, \lambda \boldsymbol{I}_P)$, where $\boldsymbol{w}, \boldsymbol{s} \in \mathcal{H}$ and $\lambda = c\exp\{(-j/b)\}$ for some $j \in \mathbb{N}$, and solve the optimization problem

$$\min_{\boldsymbol{w},\boldsymbol{s},\lambda} \breve{e}(Q) + \sqrt{\frac{\mathrm{KL}(Q\|P) + \log\frac{|S|}{\delta}}{2(|S|-1)}}, \tag{C.269}$$

with initialization $\boldsymbol{w} = \theta$, $\boldsymbol{s} = \theta^2$, one can achieved a nonvacous PAC-Bayes bound

by Equation C.268.

In order to avoid discrete optimization for $j \in \mathbb{N}$, Dziugaite and Roy (2017) uses the B_{RE} term to replace the bound in Table C.268. The B_{RE} term is defined as

$$B_{\mathrm{RE}}(\boldsymbol{w}, \boldsymbol{s}, \lambda; \delta) = \frac{\mathrm{KL}(P\|Q) + 2\log\left(b\log\frac{c}{\lambda}\right) + \log\frac{\pi^2|S|}{6\delta}}{|S| - 1}, \tag{C.270}$$

where $Q = \mathcal{N}(\boldsymbol{w}, \mathrm{diag}(\boldsymbol{s}))$, $P = \mathcal{N}(\theta_0, \lambda \boldsymbol{I}_P)$. The optimization goal actually used in the implementation is thus

$$\min_{\boldsymbol{w}\in\mathbb{R}^P, \boldsymbol{s}\in\mathbb{R}^P_+, \lambda\in(0,c)} \breve{e}(Q) + \sqrt{\frac{1}{2} B_{\mathrm{RE}}(\boldsymbol{w}, \boldsymbol{s}, \lambda; \delta)}. \tag{C.271}$$

Algorithm 4 shows the algorithm for *Iterative Hessian* (ITER) PAC-Bayes Optimization. If we set $\eta = T$, the algorithm will be come *Approximate Hessian* (APPR) PAC-Bayes Optimization. It is based on Algorithm 1 in Dziugaite and Roy (2017). The initialization of $\boldsymbol{w}$ is different from Dziugaite and Roy (2017) because we believe what they wrote, $\mathrm{abs}(\boldsymbol{w})$ is a typo and $\log[\mathrm{abs}(\boldsymbol{w})]$ is what they actually means. It is more reasonable to initialize the variance $\boldsymbol{s}$ as $\boldsymbol{w}^2$ instead of $\exp[2\,\mathrm{abs}(\boldsymbol{w})]$.

In the algorithm, HESSIANCALC($\boldsymbol{w}$) is the process to calculate Hessian information with respect to the posterior mean $\boldsymbol{w}$ in order to produce the Hessian eigenbasis to perform the change of basis. For very small networks, we can calculate Hessian explicitly but it is prohibitive for most common networks. However, efficient approximate change of basis can be performed using our approximated layer-wise Hessians. In this case, we would just need to calculate the full eigenspace of $\mathbb{E}[\boldsymbol{M}]$ and that of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ for each layer. For pth layer, we denote them as $\boldsymbol{U}^{(p)}$ and $\boldsymbol{V}^{(p)}$ respectively with eigenvectors as columns. We can also store the corresponding eigenvalues by doing pairwise multiplications between eigenvalues of $\mathbb{E}[\boldsymbol{M}]$ and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$.

Algorithm 4 PAC-Bayes bound optimization using layer-wise Hessian eigenbasis

Input:

$\boldsymbol{w}_0 \in \mathbb{R}^P$ ▷ Network parameters (Initialization)
$\boldsymbol{w} \in \mathbb{R}^P$ ▷ Network parameters (SGD solution)
S ▷ Training examples
$\delta \in (0, 1)$ ▷ Confidence parameter
$b \in \mathbb{N}, c \in (0, 1)$ ▷ Precision and bound for λ
$\tau \in (0, 1), T \in \mathbb{N}$ ▷ Learning rate; No. of iterations
$\eta \in \mathbb{N}$ ▷ Epoch interval for Hessian calculation

Output

$\boldsymbol{w}$ ▷ Optimized network parameters
$\boldsymbol{s}$ ▷ Optimized posterior variances in Hessian eigenbasis
λ ▷ Optimized prior variancce

procedure ITERATIVE-HESSIAN-PAC-BAYES
  $\boldsymbol{\varsigma} \leftarrow \log[\text{abs}(\boldsymbol{w})]$ ▷ where $\boldsymbol{s}(\boldsymbol{\varsigma}) = \exp(2\boldsymbol{\varsigma})$
  $\varrho \leftarrow -3$ ▷ where $\lambda(\varrho) = \exp(2\varrho)$
  $R(\boldsymbol{w}, \boldsymbol{s}, \lambda) = \sqrt{\frac{1}{2} B_{\text{RE}}(\boldsymbol{w}, \boldsymbol{s}, \lambda; \delta)}$ ▷ BRE term
  $B(\boldsymbol{w}, \boldsymbol{s}, \lambda, \boldsymbol{w}') = \mathcal{L}(\boldsymbol{w}') + R(\boldsymbol{w}, \boldsymbol{s}, \lambda)$ ▷ Optimization goal
  for $t = 0 \rightarrow T - 1$ **do** ▷ Run SGD for T iterations
    if $t \mod \eta == 0$ **then**
      HESSIANCALC(w)
    end if
    Sample $\boldsymbol{\xi} \sim \mathbb{N}(0, 1)^P$
    $\boldsymbol{w}'(\boldsymbol{w}, \boldsymbol{\varsigma}) = \boldsymbol{w} + \text{TOSTANDARD}\left(\boldsymbol{\xi} \odot \exp(\boldsymbol{\varsigma})\right)$ ▷ Generate noisy parameter for SNN
    $\boldsymbol{w} \leftarrow \boldsymbol{w} - \tau \left[\nabla_{\boldsymbol{w}} R(\boldsymbol{w}, \boldsymbol{s}, \lambda) + \nabla_{\boldsymbol{w}'} \mathcal{L}(\boldsymbol{w}')\right]$
    $\boldsymbol{\varsigma} \leftarrow \boldsymbol{\varsigma} - \tau \left[\nabla_{\boldsymbol{\varsigma}} R(\boldsymbol{w}, \boldsymbol{s}(\boldsymbol{\varsigma}), \lambda) + \text{TOHESSIAN}\left(\nabla_{\boldsymbol{w}'} \mathcal{L}(\boldsymbol{w}')\right) \odot \boldsymbol{\xi} \odot \exp(\boldsymbol{\varsigma})\right]$
    $\varrho \leftarrow \varrho - \tau \nabla_{\varrho} R(\boldsymbol{w}, \boldsymbol{s}, \lambda(\varrho))$ ▷ Gradient descent
  end for
  return $w, s(\boldsymbol{\varsigma}), \lambda(\varrho)$
end procedure

After getting the eigenspaces, we can perform the change of basis. Note that we perform change of basis on vectors with the same dimensionality as the parameter vector (or the posterior mean). TOHESSIAN($\boldsymbol{u}$) is the process to put a vector $\boldsymbol{u}$ in the standard basis to the Hessian eigenbasis. We first break $\boldsymbol{u}$ into different layers and let $\boldsymbol{u}^{(p)}$ be the vector for the pth layer. We then define $\text{Mat}^{(p)}$ as the reshape of a vector to the shape of the parameter matrix $\boldsymbol{W}^{(p)}$ of that layer. We have the new

vector $\boldsymbol{v}^{(p)}$ in Hessian basis as

$$\boldsymbol{v}^{(p)} = \text{vec}\left[\boldsymbol{U}^{(p)T}\,\text{Mat}^{(p)}(\boldsymbol{u}^{(p)})\boldsymbol{V}^{(p)}\right]. \tag{C.272}$$

The new vector $\boldsymbol{v} = \textsc{ToHessian}(\boldsymbol{u})$ is thus the concatenation of all the $\boldsymbol{v}^{(p)}$.

$\textsc{ToStandard}(\boldsymbol{v})$ is the process to put a vector $\boldsymbol{v}$ in the Hessian eigenbasis to the standard basis. It is the reverse process to $\textsc{ToHessian}$. We also break $\boldsymbol{v}$ into layers and let the vector for the pth layer be $\boldsymbol{v}^{(p)}$. Then, the new vector $\boldsymbol{u}^{(p)}$ is

$$\boldsymbol{u}^{(p)} = \text{vec}\left[\boldsymbol{U}^{(p)}\,\text{Mat}^{(p)}(\boldsymbol{v}^{(p)})\boldsymbol{V}^{(p)T}\right], \tag{C.273}$$

The new vector $\boldsymbol{u} = \textsc{ToStandard}(\boldsymbol{v})$ is thus the concatenation of all $\boldsymbol{u}^{(p)}$.

After getting optimized $\boldsymbol{w}, \boldsymbol{s}, \lambda$, we compute the final bound using Monte Carlo methods same as in Dziugaite and Roy (2017).

Note that the prior P is invariant with respect to the change of basis, since its covariance matrix is a multiple of identity $\lambda \boldsymbol{I}_P$. Thus, the KL divergence can be calculate in the Hessian eigenbasis without changing the value of λ. In the *Iterative Hessian with approximated output Hessian* ($\textsc{Iter.M}$), we use $\widetilde{M}$ to approximate $\mathbb{E}[\boldsymbol{M}]$, as in Equation C.265.

We followed the experiment setting proposed by Dziugaite and Roy (2017) in general. In all the results we present, we first trained the models from Gaussian random initialization w_0 to the initial posterior mean estimate w using SGD (lr=0.01) with batch-size 128 and epoch number 1000.

We then optimize the posterior mean and variance with layer-wise Hessian information using Algorithm 4, where $\delta = 0.025$, $b = 100$, and $c = 0.1$. We train for 2000 epochs, with learning rate τ initialized at 0.001 and decays with ratio 0.1 every 400 epochs. For *Approximated Hessian* algorithm, we set $\eta = 1$. For *Iterative Hessian*

algorithm, we set $\eta = 10$. We also tried η with the same decay schedule as learning rate (multiply η by 10 every time the learning rate is multiplied by 0.1) and the results are similar to those without decay. We also used the same Monte Carlo method as in Dziugaite and Roy (2017) to calculate the final PAC-Bayes bound. Except that we used 50000 iterations instead of 150000 iterations because extra iterations do not further tighten the bound significantly. We use sample frequency 100 and $\delta' = 0.01$ as in that paper.

The complete experiment results are listed in Table C.8. We follow the same naming convention as in Dziugaite and Roy (2017) except adding T-200^2 we introduced in Section 4.3. T-600_{10}, T-600^2_{10}, and T-200^2_{10} are trained on standard MNIST with 10 classes, and others are trained on MNIST-2 (see Section C.5.1), in which we combined class 0-4 and class 5-9.

In Table C.8, Prev means the previous results in Dziugaite and Roy (2017), APPR means *Approximated Hessian*, ITER means *Iterative Hessian*, ITER (D) means *Iterative Hessian* with decaying η, ITER.M means *Iterative Hessian with approximated output Hessian*. BASE are Base PAC-Bayes optimization as in the previous paper.

We also plotted the final posterior variance, $\boldsymbol{s}$. Figure C.27 shown below is for T-200^2_{10}. For posterior variance optimized with our algorithms (APPR, ITER, and ITER.M) we can see that direction associated with larger eigenvalue has a smaller variance. This agrees with our presumption that top eigenvectors are aligned with sharper directions and should have smaller variance after optimization. The effect is more significant and consistent for Iterative Hessian, where the PAC-Bayes bound is also tighter.

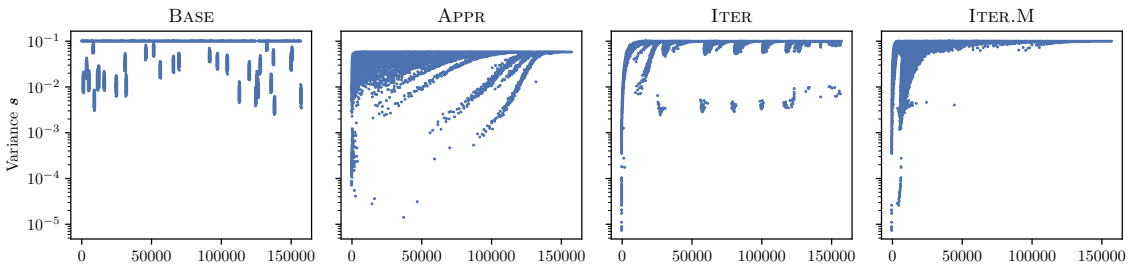
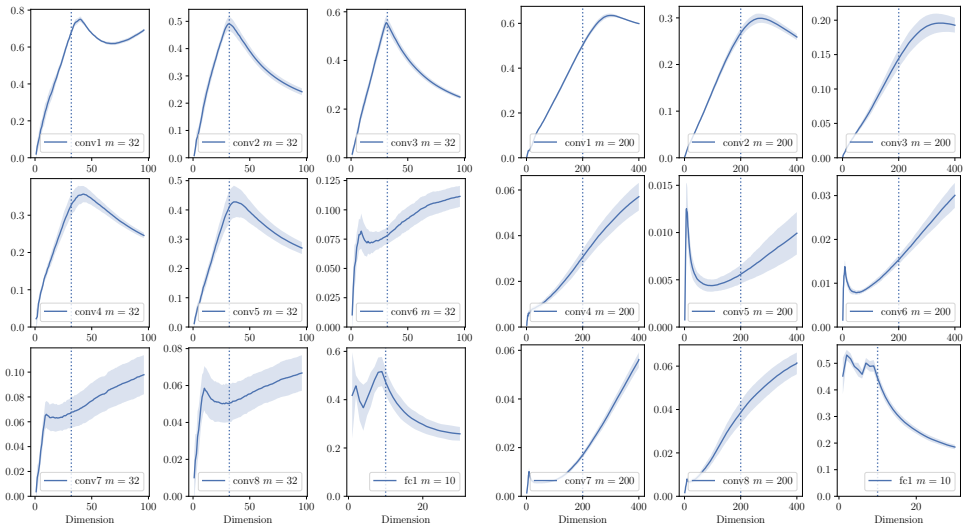


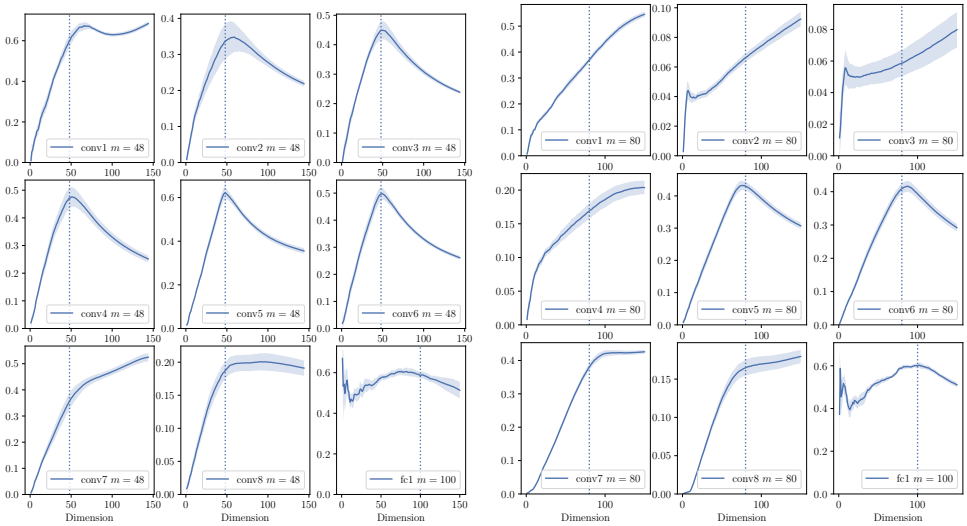
FIGURE C.27: Optimized posterior variance, $\boldsymbol{s}$. (fc1:T-200^2, trained on MNIST), the horizontal axis is ordered with decreasing eigenvalues.

Table C.8: Full PAC-Bayes bound optimization results

Network	**Method**	PAC-Bayes Bound	KL Divergence	SNN loss	λ (prior)	Test Error
T-600	PREV	0.161	5144	0.028	-	0.017
	BASE	0.154	4612.6	0.03373	-1.3313	0.0153
	APPR	0.1432	3980.6	0.03417	-1.6063	0.0153
	ITER	**0.1198**	3766.1	0.02347	-1.2913	0.0153
	ITER(D)	0.1199	3751.1	0.02366	-1.2913	0.0153
	ITER.M	0.1255	3929.9	0.02494	-1.3213	0.0153
T-600^2	PREV	0.186	6534	0.028	-	0.016
	BASE	0.1921	6966.6	0.03262	-1.4163	0.0148
	APPR	0.1658	5176.1	0.03468	-2.0963	0.0148
	ITER	0.1456	5086.5	0.02473	-1.7963	0.0148
	ITER(D)	**0.1443**	4956.8	0.02523	-1.7963	0.0148
	ITER.M	0.1502	5024.5	0.02767	-1.8363	0.0148
T-1200	PREV	0.179	5977	0.027	-	0.016
	BASE	0.1754	5917.6	0.03295	-1.5463	0.0161
	APPR	0.1725	5318.8	0.03701	-1.8313	0.0161
	ITER	0.1417	5071	0.02292	-1.4763	0.0161
	ITER(D)	**0.1413**	5021.1	0.02316	-1.4763	0.0161
	ITER.M	0.1493	5185.4	0.02576	-1.5363	0.0161
T-300^2	PREV	0.17	5791	0.027	-	0.015
	BASE	0.1686	5514.9	0.03329	-1.1513	0.015
	APPR	0.1434	4105.4	0.03296	-1.8063	0.015
	ITER	0.1249	3873.2	0.02514	-1.4763	0.015
	ITER(D)	**0.1244**	3833.7	0.02526	-1.4763	0.015
	ITER.M	0.1308	3987.2	0.02721	-1.5713	0.015
R-600	PREV	1.352	201131	0.112	-	0.501
	BASE	0.6046	1144.8	0.507	-1.8263	0.4925
	APPR	0.5653	390.25	0.5066	-2.4713	0.4925
	ITER(D)	0.5681	431.62	0.5066	-2.4513	0.4925
	ITER.M	**0.5616**	340.62	0.5065	-2.5263	0.4925
T-200^2_{10}	BASE	0.4165	21896	0.04706	-1.1513	0.0208
	APPR	0.2621	11068	0.0366	-1.4213	0.0208
	ITER	**0.2145**	9821	0.02229	-1.1513	0.0208
	ITER(D)	0.2311	9758.5	0.03071	-1.1513	0.0208
	ITER.M	0.2728	13406	0.02605	-1.1513	0.0208
T-600_{10}	BASE	0.2879	12674	0.03854	-1.1513	0.018
	APPR	0.2424	9095.8	0.04159	-1.6013	0.018
	ITER	**0.2132**	8697.9	0.02947	-1.3063	0.018
	ITER.M	0.2227	8870.9	0.03294	-1.4613	0.018
T-600^2_{10}	BASE	0.3472	17212	0.03884	-1.1513	0.0186
	APPR	0.2896	11618	0.04723	-2.0563	0.0186
	ITER	**0.2431**	10568	0.03057	-1.5713	0.0186

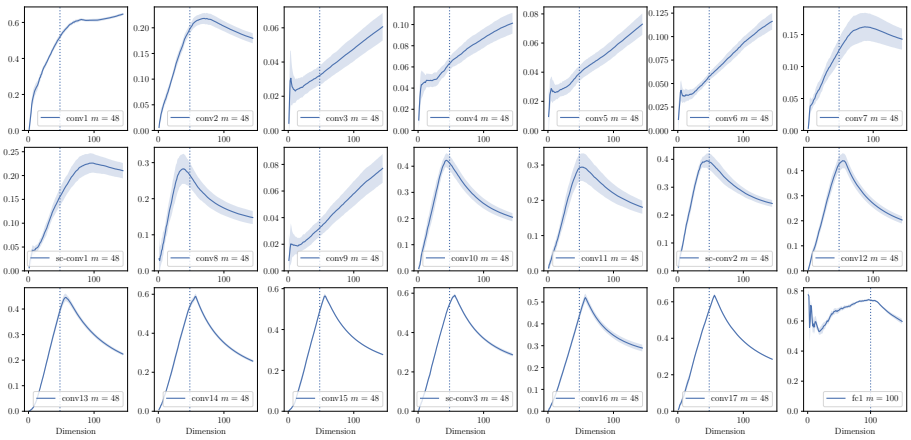


(a) VGG11-W32 (CIFAR10) (b) VGG11-W200 (CIFAR10)

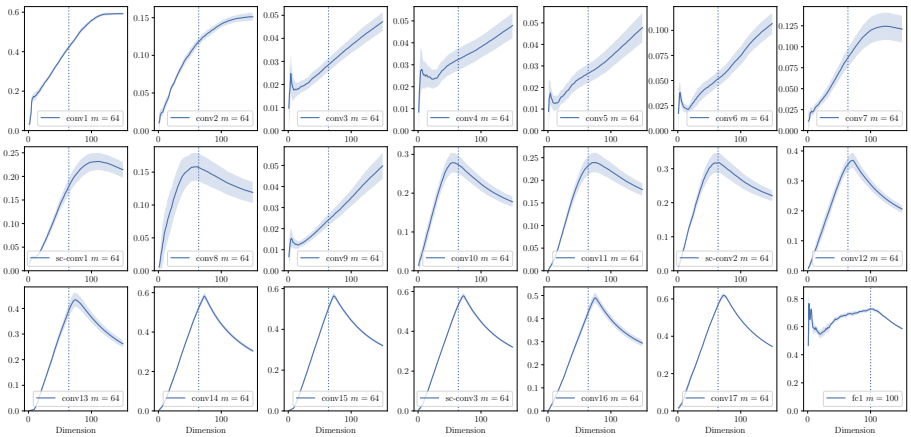


(c) VGG11-W48 (CIFAR100) (d) VGG11-W80 (CIFAR100)

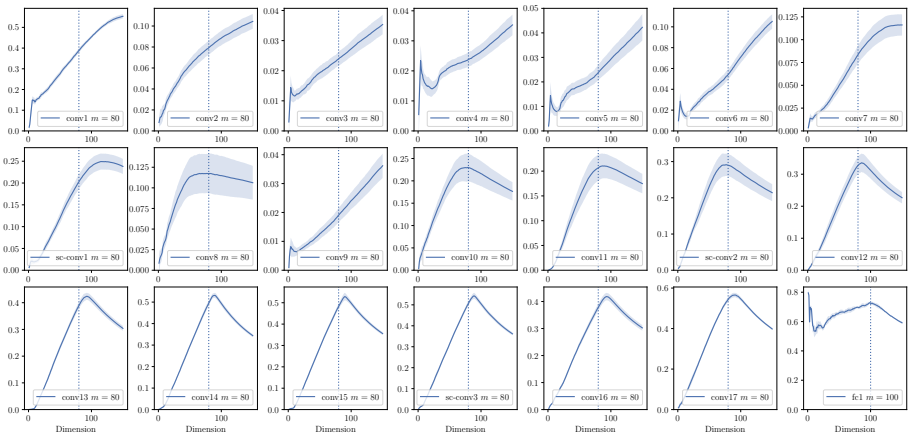
FIGURE C.3: Top Eigenspace overlap for varients of VGG11 on CIFAR10 and CIFAR100



(a) ResNet18-W48 (CIFAR100)

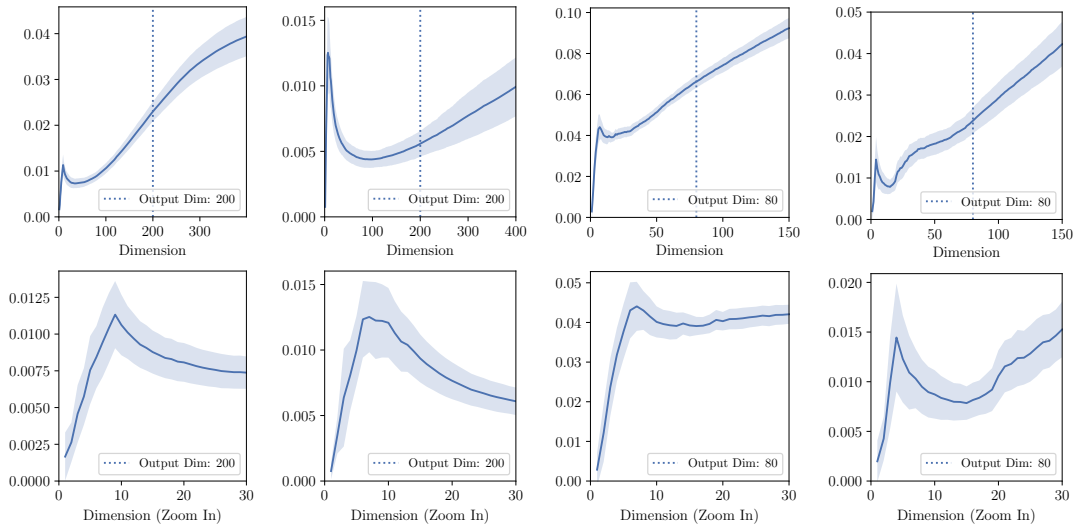


(b) ResNet18-W64 (CIFAR100)



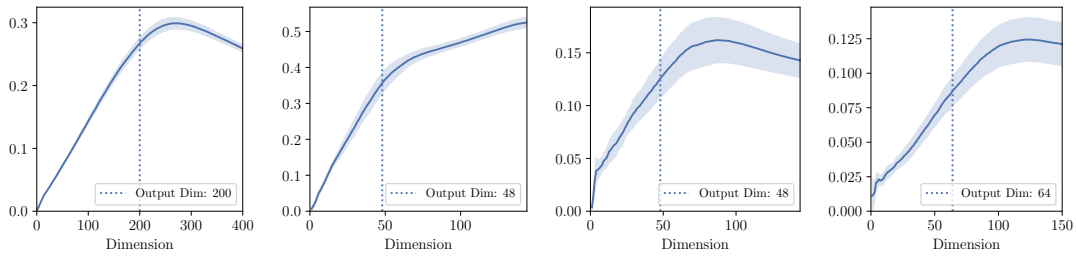
(c) ResNet18-W80 (CIFAR100)

FIGURE C.4: Top Eigenspace overlap for variants of ResNet18 on CIFAR100



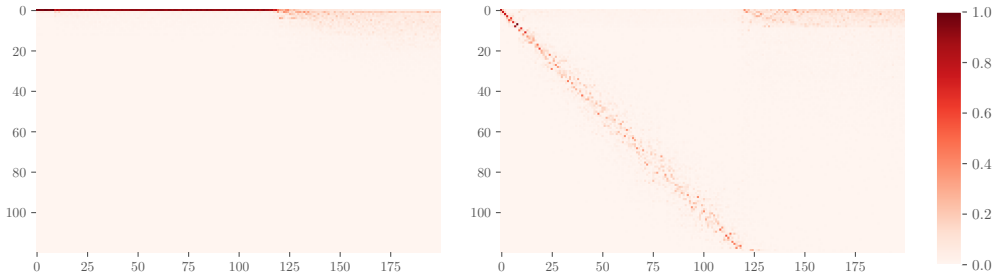
(a) fc2:F-200^2 (MNIST) (b) fc2:F-200^2 (MNIST) (c) fc2:F-200^2 (MNIST) (d) fc2:F-200^2 (MNIST)

FIGURE C.5: Top eigenspace overlap for layers with an early low peak. Figures in the second row are the zoomed in versions of the figures in the first row.

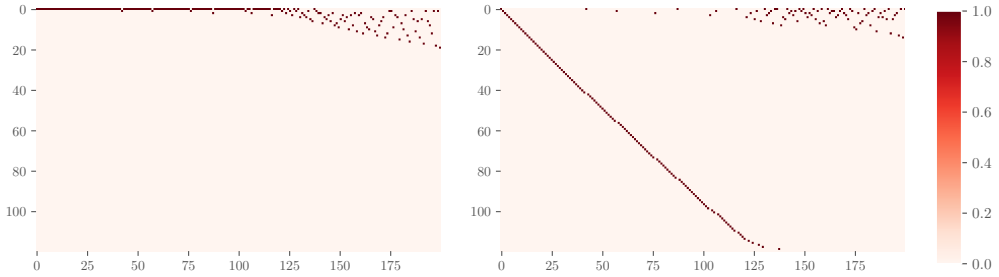


(a) conv2:VGG11-W200 (CIFAR10) (b) conv7:VGG11-W48 (CIFAR100) (c) conv7:VGG11-W48 (CIFAR100) (d) conv7:ResNet18-W64 (CIFAR100)

FIGURE C.6: Top eigenspace overlap for layers with a delayed peak.

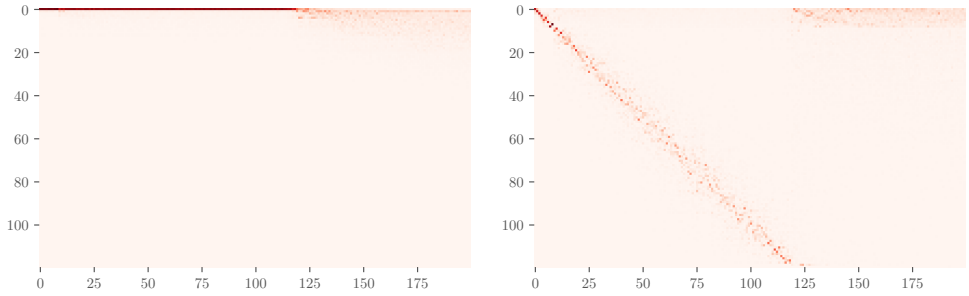


(a) True Hessian with $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right]$ (b) True Hessian with $\mathbb{E}\left[\boldsymbol{M}\right]$



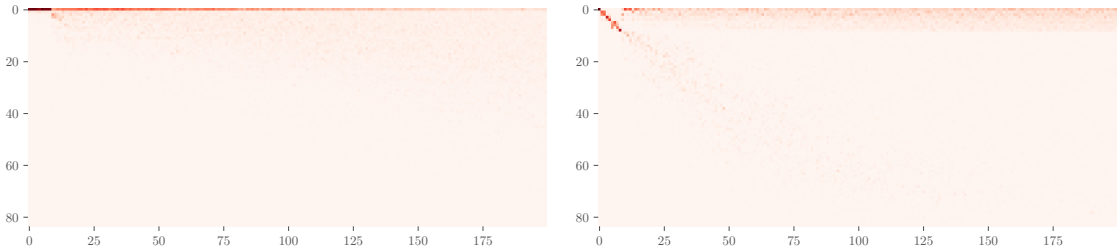
(d) Approximated Hessian with $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right]$ (e) Approximated Hessian with $\mathbb{E}\left[\boldsymbol{M}\right]$

FIGURE C.7: Heatmap of Eigenvector Correspondence Matrices for fc1:LeNet5.



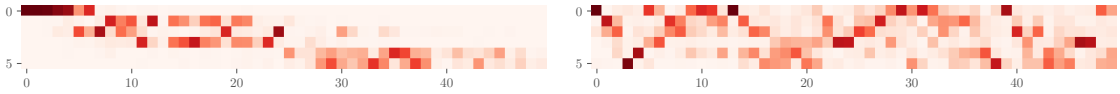
(a) Correspondence with $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right]$ (b) True Hessian with $\mathbb{E}\left[\boldsymbol{M}\right]$

FIGURE C.8: Eigenvector Correspondence for fc1:LeNet5. (m=120)



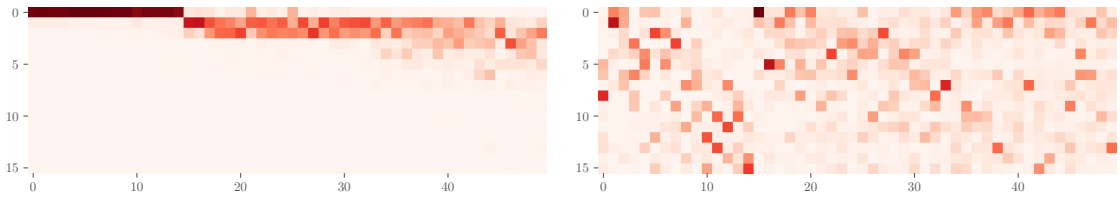
(a) Correspondence with $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right]$ (b) Correspondence with $\mathbb{E}\left[\boldsymbol{M}\right]$

FIGURE C.9: Eigenvector Correspondence for fc2:LeNet5. (m=84)



(a) Correspondence with $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right]$ (b) Correspondence with $\mathbb{E}\left[\boldsymbol{M}\right]$

FIGURE C.10: Eigenvector Correspondence for conv1:LeNet5. (m=6)



(a) Correspondence with $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right]$ (b) Correspondence with $\mathbb{E}\left[\boldsymbol{M}\right]$

FIGURE C.11: Eigenvector Correspondence for conv2:LeNet5. (m=16)

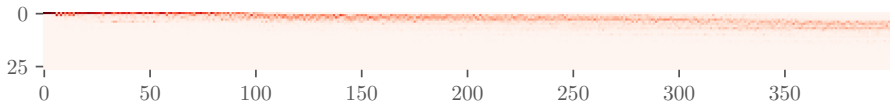


FIGURE C.12: Eigenvector Correspondence with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ for conv1:VGG11. (m=64)

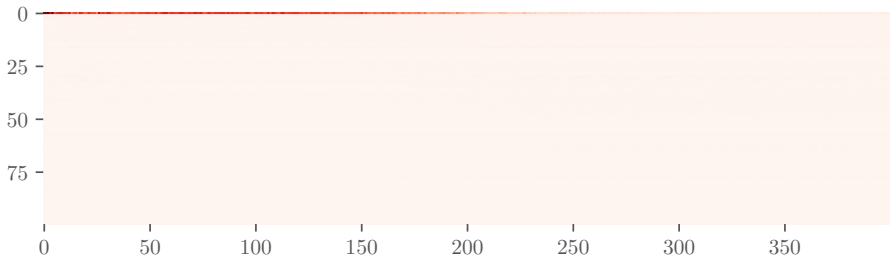


FIGURE C.13: Eigenvector Correspondence with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ for conv2:VGG11. (m=128)

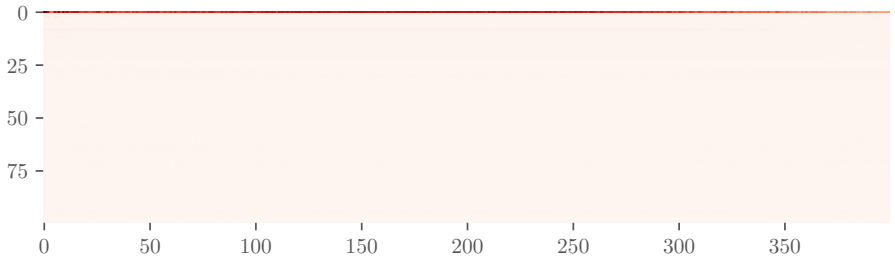


FIGURE C.14: Eigenvector Correspondence with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ for conv3:VGG11. (m=256)

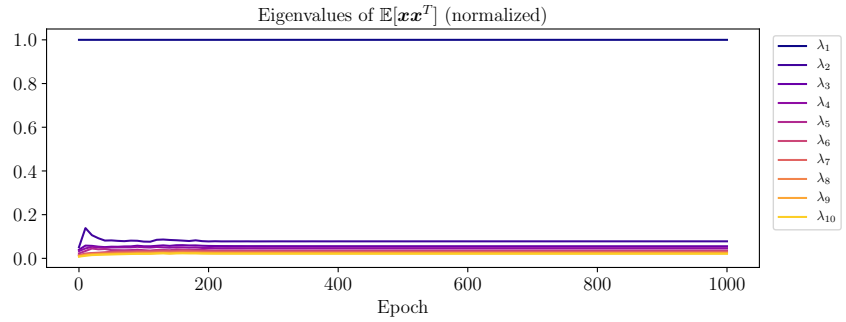


FIGURE C.15: Top eigenvalues of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ along training trajectory. (fc1:LeNet5)

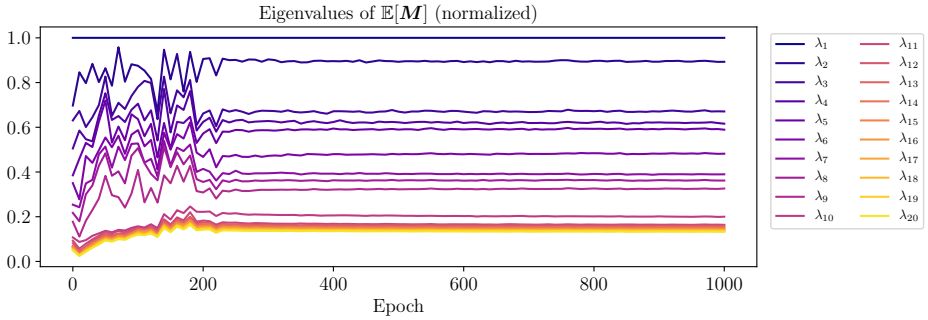


FIGURE C.16: Top eigenvalues of $\mathbb{E}[\boldsymbol{M}]$ along training trajectory. (fc1:LeNet5)

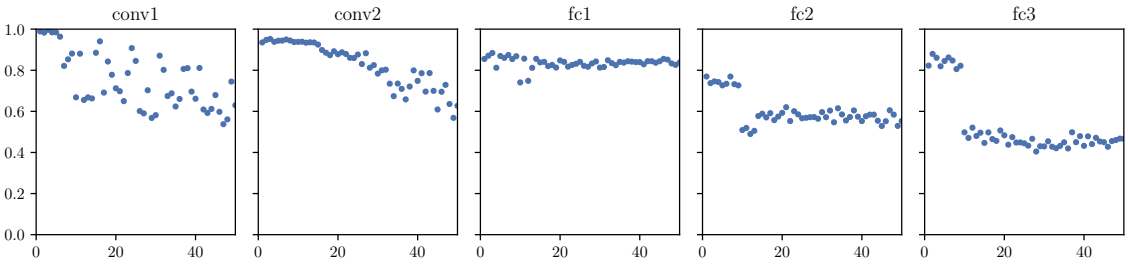
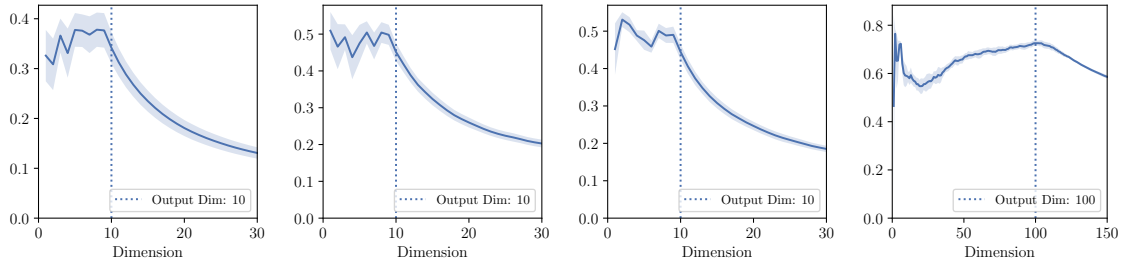
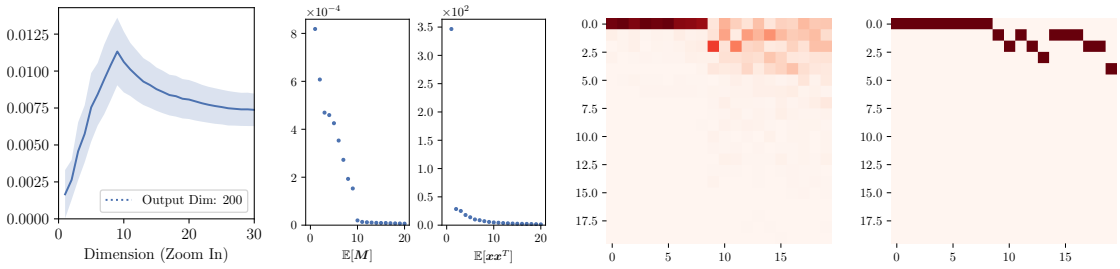


FIGURE C.17: Ratio between top singular value and Frobenius norm of matricized dominating eigenvectors. (LeNet5 on CIFAR10). The horizontal axes denote the index i of eigenvector $\boldsymbol{h}_i$, and the vertical axes denote $\|\operatorname{Mat}(\boldsymbol{h}_i)\|/\|\operatorname{Mat}(\boldsymbol{h}_i)\|_F$.



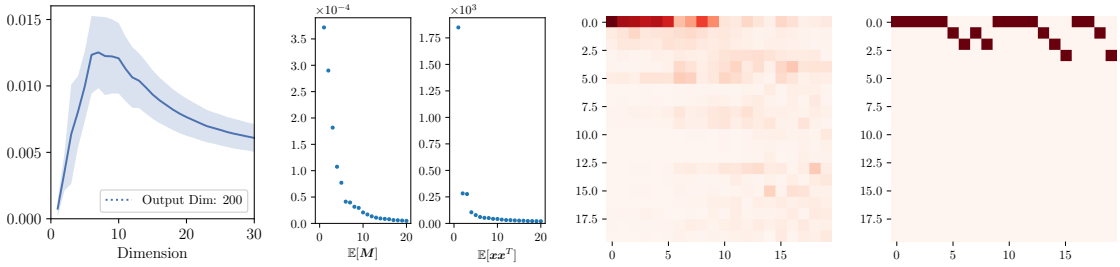
(a) fc3:F-200[2] (MNIST) (b) fc3:LeNet5 (CIFAR10) (c) fc1:VGG11-W200 (CIFAR10) (d) fc1:ResNet18-W64 (CIFAR100)

FIGURE C.18: Top eigenspace overlap for the final fully connected layer.



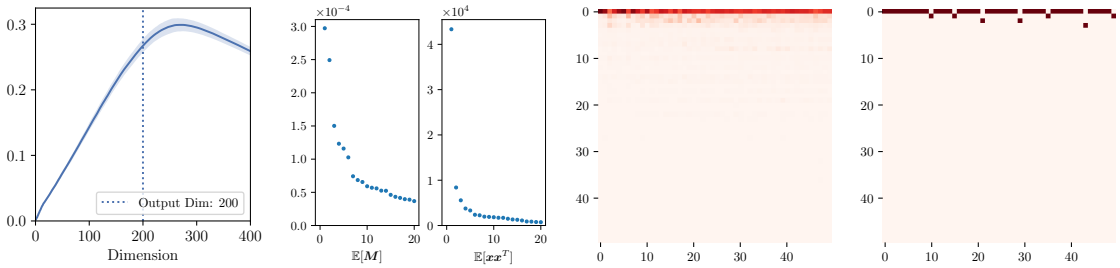
(a) Eigenspace overlap (zoomed in) (b) Eigenspectrum of $\mathbb{E}\left[\boldsymbol{M}\right]$ and $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}\right]$ (c) True Hessian with $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}\right]$ (d) Approximated Hessian with $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}\right]$

FIGURE C.19: Eigenspace overlap, eigenspectrum, and cropped (upper 20×20 block) eigenvector correspondence matrices for fc2:F-200^2 (MNIST)



(a) Eigenspace overlap (zoomed in) (b) Eigenspectrum of $\mathbb{E}\left[\boldsymbol{M}\right]$ and $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}\right]$ (c) True Hessian with $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}\right]$ (d) Approximated Hessian with $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}\right]$

FIGURE C.20: Eigenspace overlap, eigenspectrum, and cropped (upper 50×50 block) eigenvector correspondence matrices for conv2:VGG11-W200 (CIFAR10)

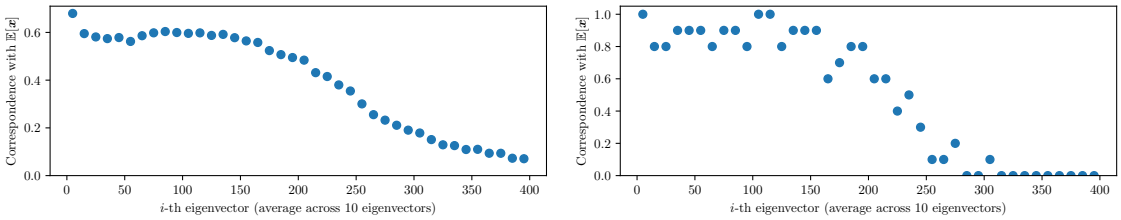


(a) Eigenspace overlap (zoomed in)

(b) Eigenspectrum of $\mathbb{E}[\boldsymbol{M}]$ and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$

(c) True Hessian with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$

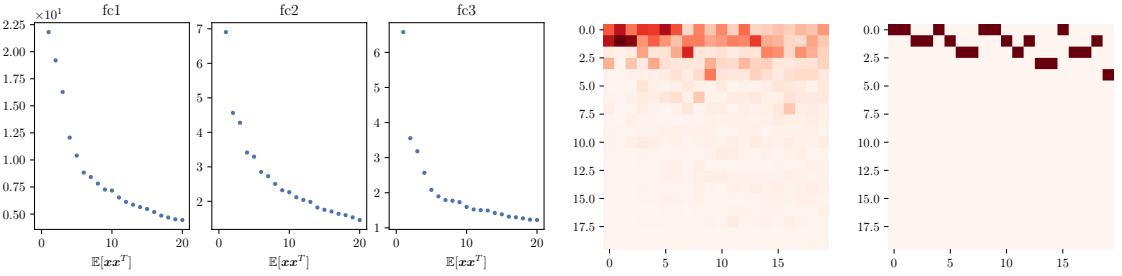
(d) Approximated Hessian with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$



(e) First Row of Correspondence Matrix of True Hessian with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$

(f) First Row of Correspondence Matrix of Approximated Hessian with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$

FIGURE C.21: Eigenspace overlap, eigenspectrum, and cropped (upper 50×50 block) eigenvector correspondence matrices for conv2:VGG11-W200 (CIFAR10)



(a) Eigenspectrum for $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$

(b) True Hessian with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ (fc1:LeNet5-BN)

(c) Approx Hessian with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ (fc1:LeNet5-BN)

FIGURE C.22: Eigenspectrum and Eigenvector correspondence matrices with $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}]$ for LeNet5-BN.

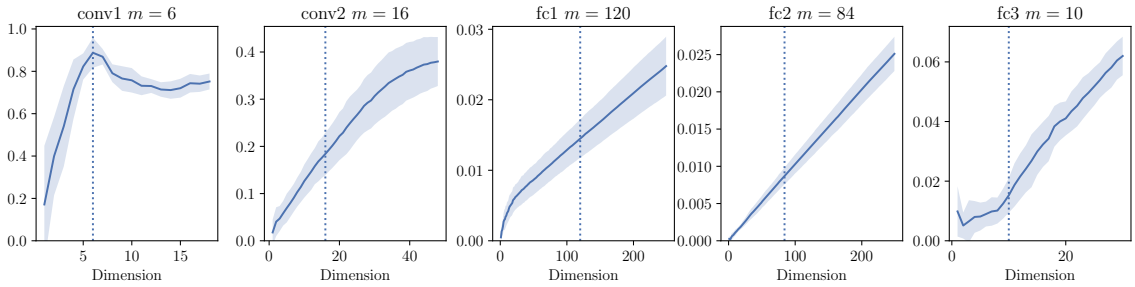
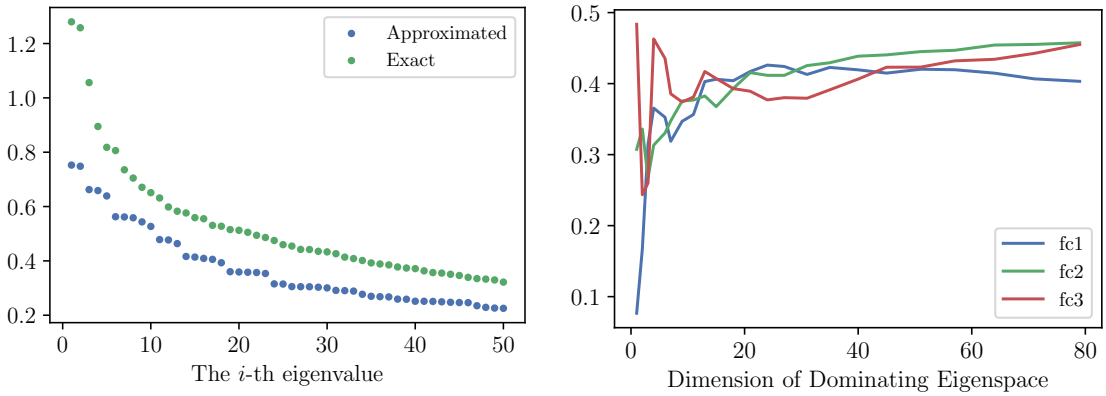


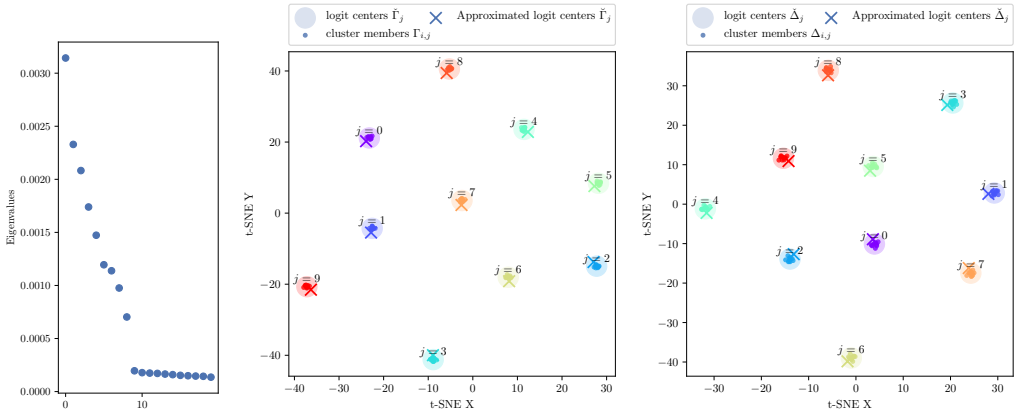
FIGURE C.23: Eigenspace overlap of different models of LeNet5-BN.



(a) Top eigenvalues of approximated and exact layer-wise Hessian for fc2

(b) Top eigenspace overlap between approximated and true layer-wise Hessian

FIGURE C.24: Comparison between the true and approximated layer-wise Hessians for LeNet5-BN.

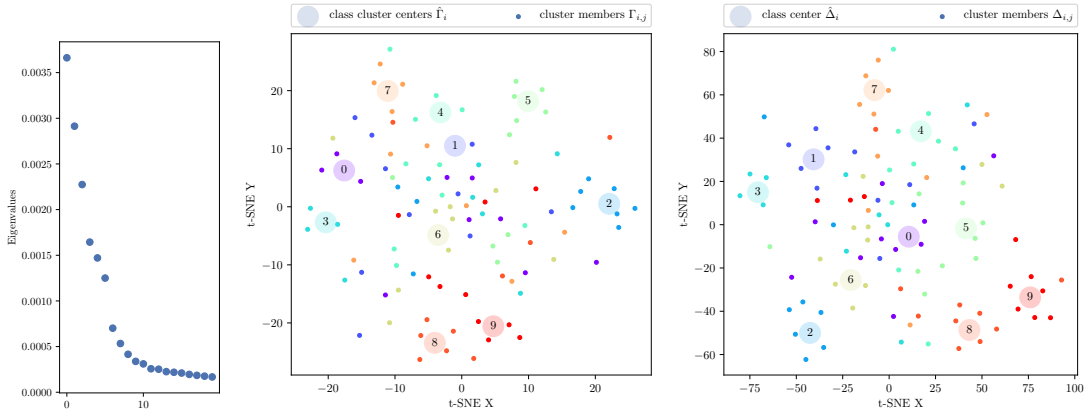


(a) Eigenspectrum of $\mathbb{E}[\boldsymbol{M}]$ at initialization

(b) Clustering of $\boldsymbol{\Gamma}$ with logits at initialization

(c) Clustering of $\boldsymbol{\Delta}$ with logits at initialization

FIGURE C.25: Logit clustering behavior of $\boldsymbol{\Delta}$ and $\boldsymbol{\Gamma}$ at initialization (fc1:T-200^2)



(a) Eigenspectrum of $\mathbb{E}[\boldsymbol{M}]$ at minimum

(b) Clustering of $\boldsymbol{\Gamma}$ with logits at minimum

(c) Clustering of $\boldsymbol{\Delta}$ with logits at minimum

FIGURE C.26: Class clustering behavior of $\boldsymbol{\Delta}$ and $\boldsymbol{\Gamma}$ at minimum. (fc1:T-200^2)

Appendix D

Supplementary Materials for Chapter 5

In this appendix section, we first give the missing proofs for the theorems in Chapter 5. Later in Appendix D.6 we give details for the experiments.

Notations: Besides the notations defined in Section 4.2, we define more notations that will be used in the proofs.

For a matrix $X \in \mathbb{R}^{n\times d}$ with $n \leqslant d$, we denote its singular values as $\sigma_1(X) \geqslant \cdots \geqslant \sigma_n(X)$.

For a positive semi-definite matrix $A \in \mathbb{R}^{d\times d}$, we denote $u^\top Au$ as $\|u\|_A^2$. For a matrix $X \in \mathbb{R}^{d\times n}$, let $\mathrm{Proj}_X \in \mathbb{R}^{d\times d}$ be the projection matrix onto the column span of X. That means, $\mathrm{Proj}_X = SS^\top$, where the columns of S form an orthonormal basis for the column span of X.

For any event $\mathcal{E}$, we use $\mathbb{1}\{\mathcal{E}\}$ to denote its indicator function: $\mathbb{1}\{\mathcal{E}\}$ equals 1 when $\mathcal{E}$ holds and equals 0 otherwise. We use $\bar{\mathcal{E}}$ to denote the complementary event of $\mathcal{E}$.

D.1 Proofs for Section 5.3 – Alleviating Gradient Explosion/Vanishing Problem for Quadratic Objective

In this section, we prove the results in Section 5.3. Recall the meta learning problem as follows:

The inner task is a fixed quadratic problem, where the starting point is fixed at w_0, and the loss function is $f(w) = \frac{1}{2} w^\top H w$ for some fixed positive definite matrix $H \in \mathbb{R}^{d \times d}$. Suppose the eigenvalue decomposition of H is $\sum_{i=1}^{d} \lambda_i u_i u_i^\top$. In this section, we assume $L = \lambda_1(H)$ and $\alpha = \lambda_d(H)$ are the largest and smallest eigenvalues of H with $L > \alpha$. We assume the starting point w_0 has unit ℓ_2 norm. For each $i \in [d]$, let c_i be $\langle w_0 \rangle u_i$ and let $c_{\min} = \min(|c_1|, |c_d|)$. We assume $c_{\min} > 0$ for simplicity, which is satisfied if w_0 is chosen randomly from the unit sphere.

Let $\{w_{\tau,\eta}\}$ be the GD sequence running on $f(w)$ starting from w_0 with step size η. For the meta-objective, we consider using the loss of the last point directly, or using the log of this value. In Section D.1.1, we first show that although choosing $\widehat{F}(\eta) = f(w_{t,\eta})$ does not have any bad local optimal solution, it has the gradient explosion/vanishing problem (Theorem 5.3.1). Then, in Section D.1.2, we show choosing $\widehat{F}(\eta) = \frac{1}{t} \log f(w_{t,\eta})$ leads to polynomially bounded meta-gradient and further show meta-gradient descent converges to the optimal step size (Theorem 5.3.2). Although the meta-gradient is polynomially bounded, if we simply use back-propogation to compute the meta-gradient, the intermediate results can still be exponentially large/small (Corollary 5.3.3). This is also proved in Section D.1.2.

D.1.1 Meta-Gradient Vanishing/Explosion

In this section, we show although choosing $\widehat{F}(\eta) = f(w_{t,\eta})$ does not have any bad local optimal solution, it has the meta-gradient explosion/vanishing problem. Recall

Theorem 5.3.1 as follows.

Theorem 5.3.1. *Let the meta-objective be $\widehat{F}(\eta) = f(w_{t,\eta})$, we know $\widehat{F}(\eta)$ is a strictly convex function in η with an unique minimizer. However, for any step size $0 < \eta < 2/L$,*

$$|\widehat{F}'(\eta)| \leqslant tL^2 \max(|1-\eta\alpha|^{2t-1}, |1-\eta L|^{2t-1});$$

for any step size $\eta > 2/L$,

$$|\widehat{F}'(\eta)| \geqslant c_1^2 L^2 t(\eta L - 1)^{2t-1} - L^2 t.$$

Intuitively, if we write $w_{t,\eta}$ in the basis of the eigen-decomposition of H, then each coordinate evolve exponentially in t. The gradient of the standard objective is therefore also exponential in t.

Proof of Theorem 5.3.1. According to the gradient descent iterations, we have

$$w_{t,\eta} = w_{t-1,\eta} - \eta \nabla f(w_{t-1,\eta}) = w_{t-1,\eta} - \eta H w_{t-1,\eta} = (I - \eta H) w_{t-1,\eta} = (I - \eta H)^t w_0.$$

Therefore, $\widehat{F}(\eta) := f(w_{t,\eta}) = \frac{1}{2} w_0^\top (I - \eta H)^{2t} H w_0$. Taking the derivative of $\widehat{F}(\eta)$,

$$\widehat{F}'(\eta) = -t w_0^\top (I - \eta H)^{2t-1} H^2 w_0 = -t \sum_{i=1}^{d} c_i^2 \lambda_i^2 (1 - \eta\lambda_i)^{2t-1},$$

where $c_i = \langle w_0 \rangle u_i$. Taking the second derivative of $F(\eta)$,

$$F''(\eta) = t(2t-1) w_0^\top (I - \eta H)^{2t-2} H^3 w_0 = t(2t-1) \sum_{i=1}^{d} c_i^2 \lambda_i^3 (1 - \eta\lambda_i)^{2t-2}.$$

Since $L > \alpha$, we have $\widehat{F}''(\eta) > 0$ for any η. That means $\widehat{F}(\eta)$ is a strictly convex function in η with a unique minimizer.

For any fixed $\eta \in (0, 2/L)$ we know $|1 - \eta\lambda_i| < 1$ for all $i \in [d]$. We have

$$\begin{aligned}\left|\widehat{F}'(\eta)\right| \leqslant & t\sum_{i=1}^{d} c_i^2\lambda_i^2|1-\eta\lambda_i|^{2t-1} \\ \leqslant & t\sum_{i=1}^{d} c_i^2 \max_{i\in[d]}\left(\lambda_i^2|1-\eta\lambda_i|^{2t-1}\right) \\ \leqslant & tL^2\max\left(|1-\eta\alpha|^{2t-1}, |1-\eta L|^{2t-1}\right),\end{aligned}$$

where the last inequality uses $\sum_{i=1}^{d} c_i^2 = 1$. Note for $\eta \in (0, 2/L)$, it's guaranteed that $|1 - \eta\lambda_i|$ takes the maximum at $|1 - \eta\alpha|$ or $|1 - \eta L|$.

For any fixed $\eta \in (2/L, \infty)$, we know $\eta L - 1 > 1$. We have

$$\begin{aligned}&\widehat{F}'(\eta) \\ =& -tc_1^2L^2(1-\eta L)^{2t-1} - t\sum_{i\neq 1:(1-\eta\lambda_i)\leqslant 0} c_i^2\lambda_i^2(1-\eta\lambda_i)^{2t-1} \\ & - t\sum_{i\neq 1:(1-\eta\lambda_i)>0} c_i^2\lambda_i^2(1-\eta\lambda_i)^{2t-1} \\ \geqslant & tc_1^2L^2(\eta L-1)^{2t-1} - t\sum_{i=1}^{d} c_i^2\lambda_i^2 \geqslant tc_1^2L^2(\eta L-1)^{2t-1} - L^2t,\end{aligned}$$

where the last inequality uses $\sum_{i=1}^{d} c_i^2 = 1$. □

D.1.2 Alleviating Meta-Gradient Vanishing/Explosion

We prove when the the meta objective is chosen as $\frac{1}{t}\log f(w_{t,\eta})$, the meta-gradient is polynomially bounded. Furthermore, we show meta-gradient descent can converge to the optimal step size within polynomial iterations. Recall Theorem 5.3.2 as follows.

Theorem 5.3.2. *Let the meta-objective be $\widehat{F}(\eta) = \frac{1}{t}\log f(w_{t,\eta})$. We know $\widehat{F}(\eta)$ has a unique minimizer η^* and $\widehat{F}'(\eta) = O\left(\frac{L^3}{c_{\min}^2\alpha(L-\alpha)}\right)$ for all $\eta \geqslant 0$. Let $\{\eta_k\}$ be the GD*

sequence running on $\widehat{F}$ *with meta step size* $\mu_k = 1/\sqrt{k}$. *Suppose the starting step size* $\eta_0 \leqslant M$. *Given any* $1/L > \epsilon > 0$, *there exists* $k' = \frac{M^6}{\epsilon^2} poly(\frac{1}{c_{\min}}, L, \frac{1}{\alpha}, \frac{1}{L-\alpha})$ *such that for all* $k \geqslant k'$, $|\eta_k - \eta^*| \leqslant \epsilon$.

When we take the log of the function value, the derivative of the function value with respect to η becomes much more stable. We will first show some structural result on $\widehat{F}(\eta)$ – it has a unqiue minimizer and the gradient is polynomially bounded. Further the gradient is only close to 0 when the point η is close to the unique minimizer. Then using such structural result we prove that meta-gradient descent converges.

Proof of Theorem 5.3.2. The proof consists of three claims. In the first claim, we show that $\widehat{F}$ has a unique minimizer and the minus meta derivative always points to the minimizer. In the second claim, we show that $\widehat{F}$ has bounded derivative. In the last claim, we show that for any η that is outside the ϵ-neighborhood of η^*, $|\widehat{F}'(\eta)|$ is lower bounded. Finally, we combine these three claims to finish the proof.

Claim D.1.1. *The meta objective* $\widehat{F}$ *has only one stationary point that is also its unique minimizer* η^*. *For any* $\eta \in [0, \eta^*)$, $\widehat{F}'(\eta) < 0$ *and for any* $\eta \in (\eta^*, \infty)$, $\widehat{F}'(\eta) > 0$. *Furthermore, we know* $\eta^* \in [1/L, 1/\alpha]$.

We can compute the derivative of $\widehat{F}$ in η as follows,

$$\widehat{F}'(\eta) = \frac{-2w_0^\top (I - \eta H)^{2t-1} H^2 w_0}{w_0^\top (I - \eta H)^{2t} H w_0} = \frac{-2\sum_{i=1}^d c_i^2 \lambda_i^2 (1-\eta\lambda_i)^{2t-1}}{\sum_{i=1}^d c_i^2 \lambda_i (1-\eta\lambda_i)^{2t}}. \tag{D.1}$$

It's not hard to verify that the denominator $\sum_{i=1}^d c_i^2 \lambda_i (1-\eta\lambda_i)^{2t}$ is always positive. Denote the numerator $-2\sum_{i=1}^d c_i^2 \lambda_i^2 (1-\eta\lambda_i)^{2t-1}$ as $g(\eta)$. Since $g'(\eta) > 0$ for any $\eta \in [0, \infty)$, we know $g(\eta)$ is strictly increasing in η. Combing with the fact that $g(0) < 0$ and $g(\infty) > 0$, we know there is a unique point (denoted as η^*) where

$g(\eta^*) = 0$ and $g(\eta) < 0$ for all $\eta \in [0, \eta^*)$ and $g(\eta) > 0$ for all $\eta \in (\eta^*, \infty)$. Since the denominator in $\widehat{F}'(\eta)$ is always positive and the numerator equals $g(\eta)$, we know there is a unique point η^* where $\widehat{F}'(\eta^*) = 0$ and $\widehat{F}'(\eta) < 0$ for all $\eta \in [0, \eta^*)$ and $\widehat{F}'(\eta) > 0$ for all $\eta \in (\eta^*, \infty)$. It's clear that η^* is the minimizer of $\widehat{F}$.

Also, it's not hard to verify that for any $\eta \in [0, 1/L)$, $\widehat{F}'(\eta) < 0$ and for any $\eta \in (1/\alpha, \infty)$, $\widehat{F}'(\eta) > 0$. This implies that $\eta^* \in [1/L, 1/\alpha]$.

Claim D.1.2. *For any $\eta \in [0, \infty)$, we have*

$$|\widehat{F}'(\eta)| \leqslant \frac{4L^3}{c_{\min}^2 \alpha(L-\alpha)} := D_{\max}.$$

For any $\eta \in [0, \frac{2}{\alpha+L}]$, we have $|1-\eta\lambda_i| \leqslant 1-\eta\alpha$ for all i. Dividing the numerator and denominator in $\widehat{F'}(\eta)$ by $(1-\eta\alpha)^{2t}$, we have

$$\left|\widehat{F}'(\eta)\right| = 2\frac{\left|\sum_{i=1}^d \frac{c_i^2\lambda_i^2}{1-\eta\alpha}\left(\frac{1-\eta\lambda_i}{1-\eta\alpha}\right)^{2t-1}\right|}{c_d^2\alpha + \sum_{i=1}^{d-1} c_i^2\lambda_i\left(\frac{1-\eta\lambda_i}{1-\eta\alpha}\right)^{2t}} \leqslant \frac{2\sum_{i=1}^d c_i^2\lambda_i^2}{c_d^2\alpha(1-\eta\alpha)}$$

$$\leqslant \frac{2(\alpha+L)\sum_{i=1}^d c_i^2\lambda_i^2}{c_d^2\alpha(L-\alpha)} \leqslant \frac{4L^3}{c_d^2\alpha(L-\alpha)},$$

where the second last inequality uses $\eta \leqslant \frac{2}{\alpha+L}$.

Similarly for any $\eta \in (\frac{2}{\alpha+L}, \infty)$, we have $|1-\eta\lambda_i| \leqslant \eta L - 1$ for all i. Dividing the numerator and denominator in $\widehat{F'}(\eta)$ by $(\eta L-1)^{2t}$, we have

$$\widehat{F}'(\eta) = 2\frac{\left|\sum_{i=1}^d \frac{c_i^2\lambda_i^2}{\eta L-1}\left(\frac{1-\eta\lambda_i}{\eta L-1}\right)^{2t-1}\right|}{c_1^2 L + \sum_{i=2}^d c_i^2\lambda_i\left(\frac{1-\eta\lambda_i}{\eta L-1}\right)^{2t}} \leqslant \frac{2\sum_{i=1}^d c_i^2\lambda_i^2}{c_1^2 L(\eta L-1)}$$

$$\leqslant \frac{2(\alpha+L)\sum_{i=1}^d c_i^2\lambda_i^2}{c_1^2 L(L-\alpha)} \leqslant \frac{4L^3}{c_1^2 L(L-\alpha)}$$

where the last inequality uses $\eta \geqslant \frac{2}{\alpha+L}$.

Overall, we know for any $\eta \geqslant 0$,

$$|\widehat{F}'(\eta)| \leqslant \frac{4L^3}{L-\alpha} \max\left(\frac{1}{c_d^2\alpha}, \frac{1}{c_1^2 L}\right) \leqslant \frac{4L^3}{c_{\min}^2 \alpha(L-\alpha)}.$$

Claim D.1.3. *Given $\widehat{M} \geqslant 2/\alpha$ and $1/L > \epsilon > 0$, for any $\eta \in [0, \eta^* - \epsilon] \cup [\eta^* + \epsilon, \widehat{M}]$, we have*

$$|F'(\eta)| \geqslant \min\left(\frac{2\epsilon c_d^2 \alpha^3}{L}, \frac{2\epsilon c_1^2 L^2}{(\widehat{M}L-1)^2}\right) \geqslant 2\epsilon c_{\min}^2 \min\left(\frac{\alpha^3}{L}, \frac{1}{\widehat{M}^2}\right) := D_{\min}(\widehat{M}).$$

If $\eta \in [0, \eta^* - \epsilon]$ and $\eta \leqslant \frac{2}{\alpha+L}$, we have

$$\begin{aligned}
\widehat{F}'(\eta) &= -2\frac{\sum_{i=1}^d c_i^2\lambda_i^2(1-\eta\lambda_i)^{2t-1}}{\sum_{i=1}^d c_i^2\lambda_i(1-\eta\lambda_i)^{2t}} \\
&= -2\frac{\sum_{i=1}^d c_i^2\lambda_i^2(1-\eta\lambda_i)^{2t-1} - \sum_{i=1}^d c_i^2\lambda_i^2(1-\eta^*\lambda_i)^{2t-1}}{\sum_{i=1}^d c_i^2\lambda_i(1-\eta\lambda_i)^{2t}},
\end{aligned}$$

where the second equality holds because $\sum_{i=1}^d c_i^2\lambda_i^2(1-\eta^*\lambda_i)^{2t-1} = 0$. For the numerator, we have

$$\begin{aligned}
\sum_{i=1}^d c_i^2\lambda_i^2(1-\eta\lambda_i)^{2t-1} - \sum_{i=1}^d c_i^2\lambda_i^2(1-\eta^*\lambda_i)^{2t-1} &\geqslant c_d^2\alpha^2\left((1-\eta\alpha)^{2t-1} - (1-\eta^*\alpha)^{2t-1}\right) \\
&\geqslant c_d^2\alpha^2\left((1-\eta\alpha)^{2t-1} - (1-\eta\alpha-\epsilon\alpha)^{2t-1}\right);
\end{aligned}$$

for the denominator, we have

$$\sum_{i=1}^d c_i^2\lambda_i(1-\eta\lambda_i)^{2t} \leqslant \left(\sum_{i=1}^d c_i^2\lambda_i\right)(1-\eta\alpha)^{2t},$$

where the second inequality holds because $|1-\eta\lambda_i| \leqslant 1-\eta\alpha$ for all i. Overall, we

have when $\eta \in [0, \eta^* - \epsilon]$ and $\eta \leqslant \frac{2}{\alpha+L}$,

$$\begin{aligned}\left|\widehat{F}'(\eta)\right| \geqslant & 2\frac{c_d^2\alpha^2\left((1-\eta\alpha)^{2t-1} - (1-\eta\alpha-\epsilon\alpha)^{2t-1}\right)}{\left(\sum_{i=1}^d c_i^2\lambda_i\right)(1-\eta\alpha)^{2t}} \\ \geqslant & \frac{2\epsilon c_d^2\alpha^3}{\left(\sum_{i=1}^d c_i^2\lambda_i\right)(1-\eta\alpha)} \geqslant \frac{2\epsilon c_d^2\alpha^3}{L},\end{aligned}$$

where the last inequality holds because $(1-\eta\alpha) \leqslant 1$ and $\sum_i^d c_i^2\lambda_i \leqslant L$.

Similarly, if $\eta \in [0, \eta^* - \epsilon]$ and $\eta \geqslant \frac{2}{\alpha+L}$, we have

$$\begin{aligned}\left|\widehat{F}'(\eta)\right| \geqslant & 2\frac{c_1^2L^2\left((1-\eta L)^{2t-1} - (1-\eta L-\epsilon L)^{2t-1}\right)}{\left(\sum_{i=1}^d c_i^2\lambda_i\right)(1-\eta L)^{2t}} \\ = & 2\frac{c_1^2L^2\left((\eta L+\epsilon L-1)^{2t-1} - (\eta L-1)^{2t-1}\right)}{\left(\sum_{i=1}^d c_i^2\lambda_i\right)(\eta L-1)^{2t}} \\ \geqslant & \frac{2\epsilon c_1^2L^3}{\left(\sum_{i=1}^d c_i^2\lambda_i\right)(\eta L-1)^2} \geqslant \frac{2\epsilon c_1^2\alpha^2L^2}{(L-\alpha)^2},\end{aligned}$$

where the last inequality holds because $\eta \leqslant \eta^* - \epsilon \leqslant 1/\alpha$ and $\sum_i^d c_i^2\lambda_i \leqslant L$.

If $\eta \in [\eta^* + \epsilon, \infty)$ and $\eta \leqslant \frac{2}{\alpha+L}$, we have

$$\begin{aligned}\left|\widehat{F}'(\eta)\right| \geqslant & 2\frac{c_d^2\alpha^2\left((1-\eta\alpha+\epsilon\alpha)^{2t-1} - (1-\eta\alpha)^{2t-1}\right)}{\left(\sum_{i=1}^d c_i^2\lambda_i\right)(1-\eta\alpha)^{2t}} \\ \geqslant & \frac{2\epsilon c_d^2\alpha^3}{L},\end{aligned}$$

If $\eta \in [\eta^* + \epsilon, \infty)$ and $\eta \geqslant \frac{2}{\alpha+L}$, we have

$$\begin{aligned}\left|\widehat{F}'(\eta)\right| \geqslant & 2\frac{c_1^2 L^2 \left((1-\eta L+\eta\epsilon)^{2t-1} - (1-\eta L)^{2t-1}\right)}{\left(\sum_{i=1}^d c_i^2\lambda_i\right)(1-\eta L)^{2t}} \\ \geqslant & \frac{2\epsilon c_1^2 L^3}{\left(\sum_{i=1}^d c_i^2\lambda_i\right)(\eta L-1)^2} \geqslant \frac{2\epsilon c_1^2 L^2}{(\widehat{M}L-1)^2},\end{aligned}$$

where the last inequality uses the assumption that $\eta \leqslant \widehat{M}$.

With the above three claims, we are ready to prove the optimization result. By Claim D.1.1, we know $F'(\eta) < 0$ for any $\eta \in [0, \eta^*)$ and $F'(\eta) > 0$ for any $\eta \in (\eta^*, \infty)$. So the opposite gradient descent always points to the minimizer.

Since $\mu_k = 1/\sqrt{k}$, when $k \geqslant k_1 := \frac{D_{\max}^2}{\epsilon^2}$ we know $\mu_k \leqslant \frac{\epsilon}{D_{\max}}$. By Claim D.1.2, we know $|\widehat{F}'(\eta)| \leqslant D_{\max}$ for all $\eta \geqslant 0$, which implies $|\mu_k \widehat{F}'(\eta)| \leqslant \epsilon$ for all $k \geqslant k_1$. That means, meta gradient descent will never overshoot the minimizer by more than ϵ when $k \geqslant k_1$. In other words, after k_1 meta iterations, once η enters the ϵ-neighborhood of η^*, it will never leave this neighborhood.

We also know that at meta iteration k_1, we have $\eta_{k_1} \leqslant \max(1/\alpha + D_{\max}, M) := \widehat{M}$. Here, $1/\alpha + D_{\max}$ comes from the case that the eta starts from the left of η^* and overshoot to the right of η^* by $D_{\max}$. Since $\eta^* \in [1/L, 1/\alpha]$, we have $|\eta_{k_1} - \eta^*| \leqslant \max(1/\alpha, 1/\alpha + D_{\max} - 1/L, M - 1/L) := R$. By Claim D.1.3, we know that $|\widehat{F}'(\eta)| \geqslant D_{\min}(\widehat{M})$ for any $\eta \in [0, \eta^* - \epsilon] \cup [\eta^* + \epsilon, \widehat{M}]$. Choosing some k_2 satisfying $\sum_{k=k_1}^{k_2} 1/\sqrt{k} \geqslant \frac{R}{D_{\min}}$, we know for any $k \geqslant k_2$, $|\eta_k - \eta^*| \leqslant \epsilon$. Plugging in all the bounds for $D_{\min}, D_{\max}$ from Claim D.1.3 and Claim D.1.2, we know there exists $k_1 = \frac{1}{\epsilon^2}\text{poly}(\frac{1}{c_{\min}}, L, \frac{1}{\alpha}, \frac{1}{L-\alpha}), k_2 = \frac{M^6}{\epsilon^2}\text{poly}(\frac{1}{c_{\min}}, L, \frac{1}{\alpha}, \frac{1}{L-\alpha})$ satisfying these conditions.

□

Next, we show although the meta-gradient is polynomailly bounded, the inter-

mediate results can still vanish or explode if we use back-propogation to compute the meta-gradient.

Corollary 5.3.3. *If we choose the meta-objective as* $\widehat{F}(\eta) = \frac{1}{t}\log f(w_{t,\eta})$*, when computing the meta-gradient using back-propagation, there are intermediate results that are exponentially large/small in number of inner-steps* t*.*

Proof of Corollary 5.3.3. This is done by direct calculation. If we use back-propagation to compute the derivative of $\frac{1}{t}\log(f(w_{t,\eta}))$, we need to first compute $\frac{\partial f(w_{t,\eta})}{\partial}\frac{1}{t}\log(f(w_{t,\eta}))$ that equals $\frac{1}{tf(w_{t,\eta})}$. Same as the analysis in Theorem 5.3.1, we can show $\frac{1}{tf(w_{t,\eta})}$ is exponentially large when $\eta < 2/L$ and is exponentially small when $\eta > 2/L$. □

D.2 Proofs of Train-by-Train v.s. Train-by-Validation (GD)

In this section, we show when the number of samples is small and when the noise level is a large constant, train-by-train overfits to the noise in training tasks while train-by-validation generalizes well. We separately prove the results for train-by-train and train-by-validation in Theorem D.2.1 and Theorem D.2.2, respectively. Then, Theorem 5.4.1 is simply a combination of Theorem D.2.1 and Theorem D.2.2.

Recall that in the train-by-train setting, each task P contains a training set S_{train} with n samples. The inner objective is defined as $\widehat{f}(w) = \frac{1}{2n}\sum_{(x,y)\in S_{\text{train}}}\left(\langle w\rangle x - y\right)^2$. Let $\{w_{\tau,\eta}\}$ be the GD sequence running on $\widehat{f}(w)$ from initialization 0 (with truncation). The meta-loss on task P is defined as the inner objective of the last point, $\Delta_{TbT(n)}(\eta, P) = \widehat{f}(w_{t,\eta}) = \frac{1}{2n}\sum_{(x,y)\in S_{\text{train}}}\left(\langle w_{t,\eta}\rangle x - y\right)^2$. The empirical meta objective $\widehat{F}_{TbT(n)}(\eta)$ is the average of the meta-loss across m different tasks. We show that under $\widehat{F}_{TbT(n)}(\eta)$, the optimal step size is a constant and the learned weight is far

from ground truth w^* on new tasks. We prove Theorem D.2.1 in Section D.2.2.

Theorem D.2.1. *Let the meta objective $\widehat{F}_{TbT(n)}(\eta)$ be as defined in Equation 5.3 with $n \in [d/4, 3d/4]$. Assume noise level σ is a large constant c_1. Assume unroll length $t \geqslant c_2$, number of training tasks $m \geqslant c_3 \log(mt)$ and dimension $d \geqslant c_4 \log(m)$ for certain constants c_2, c_3, c_4. With probability at least 0.99 in the sampling of the training tasks, we have*

$$\eta^*_{train} = \Theta(1) \text{ and } \mathbb{E}\left\|w_{t,\eta^*_{train}} - w^*\right\|^2 = \Omega(1)\sigma^2,$$

for all $\eta^*_{train} \in \arg\min_{\eta \geqslant 0} \widehat{F}_{TbT(n)}(\eta)$, where the expectation is taken over new tasks.

In Theorem D.2.1, $\Omega(1)$ is an absolute constant independent with σ. Intuitively, the reason that train-by-train performs badly in this setting is because there is a way to set the step size to a constant such that gradient descent converges very quickly to the empirical risk minimizer, therefore making the train-by-train objective very small. However, when the noise is large and the number of samples is smaller than the dimension, the empirical risk minimizer (ERM) overfits to the noise and is not the best solution.

In the train-by-validation setting, each task P contains a training set S_{train} with n_1 samples and a validation set with n_2 samples. The inner objective is defined as $\widehat{f}(w) = \frac{1}{2n_1} \sum_{(x,y) \in S_{\text{train}}} (\langle w \rangle x - y)^2$. Let $\{w_{\tau,\eta}\}$ be the GD sequence running on $\widehat{f}(w)$ from initialization 0 (with truncation). For each task P, the meta-loss $\Delta_{TbV(n_1,n_2)}(\eta, P)$ is defined as the loss of the last point $w_{t,\eta}$ evaluated on the validation set S_{valid}. That is, $\Delta_{TbV(n_1,n_2)}(\eta, P) = \frac{1}{2n_2} \sum_{(x,y) \in S_{\text{valid}}} (\langle w_{t,\eta} \rangle x - y)^2$. The empirical meta objective $\widehat{F}_{TbV(n_1,n_2)}(\eta)$ is the average of the meta-loss across m different tasks $P_1, P_2, ..., P_m$. We show that under $\widehat{F}_{TbV(n_1,n_2)}(\eta)$, the optimal step size

is $\Theta(1/t)$ and the learned weight is better than initialization 0 by a constant on new tasks. Theorem D.2.2 is proved in Section D.2.3.

Theorem D.2.2. *Let the meta objective $\widehat{F}_{TbV(n_1,n_2)}(\eta)$ be as defined in Equation 5.4 with $n_1, n_2 \in [d/4, 3d/4]$. Assume noise level σ is a large constant c_1. Assume unroll length $t \geqslant c_2$, number of training tasks $m \geqslant c_3$ and dimension $d \geqslant c_4 \log(t)$ for certain constants c_2, c_3, c_4. With probability at least 0.99 in the sampling of training tasks, we have*

$$\eta^*_{valid} = \Theta(1/t) \text{ and } \mathbb{E}\left\| w_{t,\eta^*_{valid}} - w^* \right\|^2 = \|w^*\|^2 - \Omega(1)$$

for all $\eta^*_{valid} \in \arg\min_{\eta \geqslant 0} \widehat{F}_{TbV(n_1,n_2)}(\eta)$, where the expectation is taken over new tasks.

Intuitively, train-by-validation is optimizing the right objective. As long as the meta-training problem has good generalization performance (that is, good performance on a few tasks implies good performance on the distribution of tasks), then train-by-validation should be able to choose the optimal learning rate. The step size of $\Theta(1/t)$ here serves as regularization similar to early-stopping, which allows gradient descent algorithm to achieve better error on test data.

Notations We define more quantities that are useful in the analysis. In the train by train setting, given a task $P_k := (\mathcal{D}(w_k^*), S_{\text{train}}^{(k)}, \ell)$. The training set $S_{\text{train}}^{(k)}$ contains n samples $\{x_i^{(k)}, y_i^{(k)}\}_{i=1}^n$ with $y_i^{(k)} = \langle w_k^* \rangle x_i^{(k)} + \xi_i^{(k)}$.

Let $X_{\text{train}}^{(k)}$ be an $n \times d$ matrix with its i-th row as $(x_i^{(k)})^\top$. We also let $H_{\text{train}}^{(k)} := \frac{1}{n}(X_{\text{train}}^{(k)})^\top X_{\text{train}}^{(k)}$ be the covariance matrix of the inputs in $S_{\text{train}}^{(k)}$. Let $\xi_{\text{train}}^{(k)}$ be an n-dimensional column vector with its i-th entry equal to $\xi_i^{(k)}$.

Since $n \leqslant d$, with probability 1, we know $X_{\text{train}}^{(k)}$ is full row rank. Therefore, $X_{\text{train}}^{(k)}$ has pseudo-inverse $(X_{\text{train}}^{(k)})^\dagger$ such that $X_{\text{train}}^{(k)}(X_{\text{train}}^{(k)})^\dagger = I_n$. It's not hard to verify that

there exists $w_{\text{train}}^{(k)} = \text{Proj}_{(X_{\text{train}}^{(k)})^\top} w_k^* + (X_{\text{train}}^{(k)})^\dagger \xi_{\text{train}}^{(k)}$ such that $y_i^{(k)} = \left\langle w_{\text{train}}^{(k)} \right\rangle x_i^{(k)}$ for every $(x_i^{(k)}, y_i^{(k)}) \in S_{\text{train}}^{(k)}$. Here, $\text{Proj}_{(X_{\text{train}}^{(k)})^\top}$ is the projection matrix onto the column span of $(X_{\text{train}}^{(k)})^\top$. We also denote $\text{Proj}_{(X_{\text{train}}^{(k)})^\top} w_k^*$ as $(w_{\text{train}}^{(k)})^*$. We use $B_{t,\eta}^{(k)}$ to denote $(I - (I - \eta H_{\text{train}}^{(k)})^t)$. Let $w_{t,\eta}^{(k)}$ be the weight obtained by running GD on $S_{\text{train}}^{(k)}$ with step size η (with truncation).

With the above notations, it's not hard to verify that for task P_k, the inner objective $\widehat{f}(w) = \frac{1}{2}\left\| w - w_{\text{train}}^{(k)} \right\|_{H_{\text{train}}^{(k)}}^2$. The meta-loss on task P_k is just $\Delta_{TbT(n)}(\eta, P_k) = \frac{1}{2}\left\| w_{t,\eta} - w_{\text{train}}^{(k)} \right\|_{H_{\text{train}}^{(k)}}^2$.

In the train-by-validation setting, each task P_k contains a training set $S_{\text{train}}^{(k)}$ with n_1 samples and a validation set $S_{\text{valid}}^{(k)}$ with n_2 samples. Similar as above, for the training set $S_{\text{train}}^{(k)}$, we can define $\xi_{\text{train}}^{(k)}, X_{\text{train}}^{(k)}, H_{\text{train}}^{(k)}, w_{\text{train}}^{(k)}, B_{t,\eta}^{(k)}, w_{t,\eta}^{(k)}$; for the validation set $S_{\text{valid}}^{(k)}$, we can define $\xi_{\text{valid}}^{(k)}, X_{\text{valid}}^{(k)}, H_{\text{valid}}^{(k)}, w_{\text{valid}}^{(k)}$. With these notations, the inner objective is $\widehat{f}(w) = \frac{1}{2}\left\| w - w_{\text{train}}^{(k)} \right\|_{H_{\text{train}}^{(k)}}^2$ and the meta-loss is $\Delta_{TbV(n_1,n_2)}(\eta, P_k) = \frac{1}{2}\left\| w_{t,\eta} - w_{\text{valid}}^{(k)} \right\|_{H_{\text{valid}}^{(k)}}^2$.

We also use these notations without index k to refer to the quantities defined on task P. In the proofs, we ignore the subsripts on n, n_1, n_2 and simply write $\Delta_{TbT}(\eta, P_k), \Delta_{TbV}(\eta, P_k), \widehat{F}_{TbT}, \widehat{F}_{TbV}, F_{TbT}, F_{TbV}$.

D.2.1 Overall Proof Strategy

In this section (and the next), we follow similar proof strategies that consists of three steps.

Step 1: First, we show for both train-by-train and train-by-validation, there is a good step size that achieves small empirical meta-objective (however the step sizes and the empirical meta-objective they achieve are different in the two settings). This does not necessarily mean that the actual optimal step size is exactly the good step size that we propose, but it gives an upperbound on the empirical meta-objective for the optimal step size.

Step 2: Second, we define a threshold step size such that for any step size larger than it, the empirical meta-objective must be higher than what was achieved at the good step size in Step 1. This immediately implies that the optimal step size cannot exceed this threshold step size.

Step 3: Third, we show the meta-learning problem has good generalization performance, that is, if a learning rate η performs well on the training tasks, it must also perform well on the task distribution, and vice versa. Thanks to Step 1 and Step 2, we know the optimal step size cannot exceed certain threshold and then only need to prove generalization result within this range. The generalization result is not surprising as we only have a single trainable parameter η, however we also emphasize that this is non-trivial as we will not restrict the step size η to be small enough that the algorithms do not diverge. Instead we use a truncation to alleviate the diverging problem (this allows us to run the algorithm on distribution of data whose largest possible learning rate is unknown).

Combing Step 1, 2, 3, we know the population meta-objective has to be small at the optimal step size. Finally, we show that as long as the population meta-objective is small, the performance of the algorithms satisfy what we stated in Theorem 5.4.1. The last step is easier for the train-by-validation setting, because its meta-objective

is exactly the correct measure that we are looking at; for the train-by-train setting we instead look at the property of empirical risk minimizer (ERM), and show that anything close to the ERM is going to behave similarly.

D.2.2 Train-by-Train (GD)

Recall Theorem D.2.1 as follows.

Theorem D.2.1. *Let the meta objective $\widehat{F}_{TbT(n)}(\eta)$ be as defined in Equation 5.3 with $n \in [d/4, 3d/4]$. Assume noise level σ is a large constant c_1. Assume unroll length $t \geqslant c_2$, number of training tasks $m \geqslant c_3 \log(mt)$ and dimension $d \geqslant c_4 \log(m)$ for certain constants c_2, c_3, c_4. With probability at least 0.99 in the sampling of the training tasks, we have*

$$\eta^*_{train} = \Theta(1) \text{ and } \mathbb{E}\left\|w_{t,\eta^*_{train}} - w^*\right\|^2 = \Omega(1)\sigma^2,$$

for all $\eta^*_{train} \in \arg\min_{\eta \geqslant 0} \widehat{F}_{TbT(n)}(\eta)$, where the expectation is taken over new tasks.

According to the data distribution, we know X_{train} is an $n \times d$ random matrix with each entry i.i.d. sampled from standard Gaussian distribution. In the following lemma, we show that the covariance matrix H_{train} is approximately isotropic when $d/4 \leqslant n \leqslant 3d/4$. Specifically, we show $\frac{\sqrt{d}}{\sqrt{L}} \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{Ld}$ and $\frac{1}{L} \leqslant \lambda_i(H_{\text{train}}) \leqslant L$ for all $i \in [n]$ with $L = 100$. We use letter L to denote the upper bound of $\|H_{\text{train}}\|$ to emphasize that this bounds the smoothness of the inner objective. Throughout this section, we use letter L to denote constant 100. The proof of Lemma D.2.3 follows from random matrix theory. We defer its proof into Section D.2.2.

Lemma D.2.3. *Let $X \in \mathbb{R}^{n \times d}$ be a random matrix with each entry i.i.d. sampled from standard Gaussian distribution. Let $H := 1/nX^\top X$. Assume $n = cd$ with*

$c \in [\frac{1}{4}, \frac{3}{4}]$. *Then, with probability at least* $1-\exp(-\Omega(d))$, *there exists constant* $L = 100$ *such that*

$$\frac{\sqrt{d}}{\sqrt{L}} \leqslant \sigma_i(X) \leqslant \sqrt{Ld} \text{ and } \frac{1}{L} \leqslant \lambda_i(H) \leqslant L,$$

for all $i \in [n]$.

In this section, we always assume the size of each training set is within $[d/4, 3d/4]$ so Lemma D.2.3 holds. Since $\|H_{\text{train}}\|$ is upper bounded by L with high probability, we know the GD sequence converges to w_{train} for $\eta \in [0, 1/L]$. In Lemma 5.4.3, we prove that the empirical meta objective $\widehat{F}_{TbT}$ monotonically decreases as η increases until $1/L$. Also, we show $\widehat{F}_{TbT}$ is exponentially small in t at step size $1/L$. This serves as step 1 in Section D.2.1. The proof is deferred into Section D.2.2.

Lemma 5.4.3. *With probability at least* $1-m\exp(-\Omega(d))$, $\widehat{F}_{TbT}(\eta)$ *is monotonically decreasing in* $[0, 1/L]$ *and*

$$\widehat{F}_{TbT}(1/L) \leqslant 2L^2\sigma^2\left(1 - \frac{1}{L^2}\right)^t.$$

When the step size is larger than $1/L$, the GD sequence can diverge, which incurs a high loss in meta objective. Later in Definition D.2.5, we define a step size $\widetilde{\eta}$ such that the GD sequence gets truncated with descent probability for any step size that is larger than $\widetilde{\eta}$. In Lemma 5.4.4, we show with high probability, the empirical meta objective is high for all $\eta > \widetilde{\eta}$. This serves as step 2 in the proof strategy described in Section D.2.1. The proof is deferred into Section D.2.2.

Lemma 5.4.4. *With probability at least* $1 - \exp(-\Omega(m))$,

$$\widehat{F}_{TbT}(\eta) \geqslant \frac{\sigma^2}{10L^8}$$

for all $\eta > \widetilde{\eta}$.

By Lemma 5.4.3 and Lemma 5.4.4, we know the optimal step size must lie in $[1/L, \widetilde{\eta}]$. We can also show $1/L < \widetilde{\eta} < 3L$, so η^*_{train} is a constant. To relate the empirical loss at η^*_{train} to the population loss. We prove a generalization result for step sizes within $[1/L, \widetilde{\eta}]$. The following lemma is a formal version of Lemma 5.4.5. This serves as step 3 in Section D.2.1. The proof is deferred into Section D.2.2.

Lemma D.2.4. *Suppose* σ *is a large constant* c_1*. Assume* $t \geqslant c_2, d \geqslant c_4$ *for certain constants* c_2, c_4*. With probability at least* $1 - m\exp(-\Omega(d)) - O(t+m)\exp(-\Omega(m))$,

$$|F_{TbT}(\eta) - \widehat{F}_{TbT}(\eta)| \leqslant \frac{\sigma^2}{L^3},$$

for all $\eta \in [1/L, \widetilde{\eta}]$,

Combining the above lemmas, we know the population meta objective F_{TbT} is small at η^*_{train}, which means $w_{t,\eta^*_{\text{train}}}$ is close to the ERM solution. Since the ERM solution overfits to the noise in training tasks, we know $\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|$ has to be large. We present the proof of Theorem D.2.1 as follows.

Proof of Theorem D.2.1. We assume σ is a large constant in this proof. According to Lemma 5.4.3, we know with probability at least $1 - m\exp(-\Omega(d))$, $\widehat{F}_{TbT}(\eta)$ is monotonically decreasing in $[0, 1/L]$ and $\widehat{F}_{TbT}(1/L) \leqslant 2L^2\sigma^2(1-1/L^2)^t$. This implies that the optimal step size $\eta^*_{\text{train}} \geqslant 1/L$ and $\widehat{F}_{TbT}(\eta^*_{\text{train}}) \leqslant 2L^2\sigma^2(1-1/L^2)^t$. By Lemma 5.4.4, we know with probability at least $1 - \exp(-\Omega(m))$, $\widehat{F}_{TbT}(\eta) \geqslant \frac{\sigma^2}{10L^8}$ for all $\eta > \widetilde{\eta}$, where $\widetilde{\eta}$ is defined in Definition D.2.5. As long as $t \geqslant c_2$ for certain constant c_2, we know $\frac{\sigma^2}{10L^8} > 2L^2\sigma^2(1-1/L^2)^t$, which then implies that the optimal step size η^*_{train} lies in $[1/L, \widetilde{\eta}]$. According to Lemma D.2.7, we know $\widetilde{\eta} \in (1/L, 3L)$. Therefore η^*_{train} is a constant.

According to Lemma D.2.4, we know with probability at least $1-m\exp(-\Omega(d))-O(t+m)\exp(-\Omega(m))$, $|F_{TbT}(\eta)-\widehat{F}_{TbT}(\eta)|\leqslant\frac{\sigma^2}{L^3}$, for all $\eta\in[1/L,\widetilde{\eta}]$. As long as t is larger than some constant, we have $\widehat{F}_{TbT}(\eta^*_{\text{train}})\leqslant\frac{\sigma^2}{L^3}$. Combing with the generalization result, we have $F_{TbT}(\eta^*_{\text{train}})\leqslant\frac{2\sigma^2}{L^3}$. Next, we show that under a small population loss, $\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}}-w^*\right\|^2$ has to be large.

Let $\mathcal{E}_1$ be the event that $\sqrt{d}/\sqrt{L}\leqslant\sigma_i(X_{\text{train}})\leqslant\sqrt{Ld}$ and $1/L\leqslant\lambda_i(H_{\text{train}})\leqslant L$ for all $i\in[n]$ and $\sqrt{d}\sigma/4\leqslant\|\xi_{\text{train}}\|\leqslant\sqrt{d}\sigma$. We have

$$\begin{aligned}\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}}-w_{\text{train}}\right\|^2_{H_{\text{train}}}\geqslant&\frac{1}{L}\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}}-w_{\text{train}}\right\|^2\mathbb{1}\left\{\mathcal{E}_1\right\}\\\geqslant&\frac{1}{L}\left(\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}}-w^*_{\text{train}}-(X_{\text{train}})^\dagger\xi_{\text{train}}\right\|\mathbb{1}\left\{\mathcal{E}_1\right\}\right)^2\\\geqslant&\frac{1}{L}\left(\mathbb{E}\left\|(X_{\text{train}})^\dagger\xi_{\text{train}}\right\|\mathbb{1}\left\{\mathcal{E}_1\right\}-\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}}-w^*_{\text{train}}\right\|\mathbb{1}\left\{\mathcal{E}_1\right\}\right)^2.\end{aligned}$$

Since $\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}}-w_{\text{train}}\right\|^2_{H_{\text{train}}}\leqslant\frac{4\sigma^2}{L^3}$, this then implies

$$\mathbb{E}\left\|(X_{\text{train}})^\dagger\xi_{\text{train}}\right\|\mathbb{1}\left\{\mathcal{E}_1\right\}-\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}}-w^*_{\text{train}}\right\|\mathbb{1}\left\{\mathcal{E}_1\right\}\leqslant\sqrt{L\frac{4\sigma^2}{L^3}}=\frac{2\sigma}{L}.$$

Conditioning on $\mathcal{E}_1$, we can lower bound $\left\|(X_{\text{train}})^\dagger\xi_{\text{train}}\right\|$ by $\frac{\sigma}{4\sqrt{L}}$. According to Lemma D.2.3 and Lemma D.5.1, we know $\Pr[\mathcal{E}_1]\geqslant1-\exp(-\Omega(d))$. As long as d is at least certain constant, we have $\Pr[\mathcal{E}_1]\geqslant0.9$. This then implies $\mathbb{E}\left\|(X_{\text{train}})^\dagger\xi_{\text{train}}\right\|\mathbb{1}\left\{\mathcal{E}_1\right\}\geqslant\frac{9\sigma}{40\sqrt{L}}$. Therefore, we have

$$\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}}-w^*_{\text{train}}\right\|\mathbb{1}\left\{\mathcal{E}_1\right\}\geqslant\frac{9\sigma}{40\sqrt{L}}-\frac{2\sigma}{L}=\frac{9\sigma}{4L}-\frac{2\sigma}{L}=\frac{\sigma}{4L},$$

where the first equality uses $L = 100$. Then, we have

$$\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2 \geqslant \mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*_{\text{train}}\right\|^2 \mathbb{1}\left\{\mathcal{E}_1\right\}$$

$$\geqslant \left(\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*_{\text{train}}\right\| \mathbb{1}\left\{\mathcal{E}_1\right\}\right)^2 \geqslant \frac{\sigma^2}{16L^2},$$

where the first inequality holds because for any S_{train}, w^*_{train} is the projection of w^* on the subspace of S_{train} and $w_{t,\eta^*_{\text{train}}}$ is also in this subspace. Taking a union bound for all the bad events, we know this result holds with probability at least 0.99 as long as σ is a large constant c_1 and $t \geqslant c_2, m \geqslant c_3 \log(mt)$ and $d \geqslant c_4 \log(m)$ for certain constants c_2, c_3, c_4. □

Behavior of $\widehat{F}_{TbT}$ for $\eta \in [0, 1/L]$

In this section, we prove the empirical meta objective $\widehat{F}_{TbT}$ is monotonically decreasing in $[0, 1/L]$. Furthermore, we show $\widehat{F}_{TbT}(1/L)$ is exponentially small in t.

Lemma 5.4.3. *With probability at least $1 - m\exp(-\Omega(d))$, $\widehat{F}_{TbT}(\eta)$ is monotonically decreasing in $[0, 1/L]$ and*

$$\widehat{F}_{TbT}(1/L) \leqslant 2L^2\sigma^2\left(1 - \frac{1}{L^2}\right)^t.$$

Proof of Lemma 5.4.3. For each $k \in [m]$, let $\mathcal{E}_k$ be the event that $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\text{train}}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{\text{train}}\| \leqslant \sqrt{d}\sigma$. Here, L is constant 100 from Lemma D.2.3. According to Lemma D.2.3 and Lemma D.5.1, we know for each $k \in [m]$, $\mathcal{E}_k$ happens with probability at least $1 - \exp(-\Omega(d))$. Taking a union bound over all $k \in [m]$, we know $\cap_{k\in[m]}\mathcal{E}_k$ holds with probability at least $1 - m\exp(-\Omega(d))$. From now on, we assume $\cap_{k\in[m]}\mathcal{E}_k$ holds.

Let's first consider each individual loss function $\Delta_{TbT}(\eta, P_k)$. Let $\{\widehat{w}^{(k)}_{\tau,\eta}\}$ be the

GD sequence without truncation. We have

$$\begin{aligned}\widehat{w}_{\tau,\eta}^{(k)} - w_{\text{train}}^{(k)} =& \widehat{w}_{\tau-1,\eta}^{(k)} - w_{\text{train}}^{(k)} - \eta H_{\text{train}}^{(k)}(\widehat{w}_{\tau-1,\eta}^{(k)} - w_{\text{train}}^{(k)}) \\ =& (I - \eta H_{\text{train}}^{(k)})(\widehat{w}_{\tau-1,\eta}^{(k)} - w_{\text{train}}^{(k)}) = -(I - \eta H_{\text{train}}^{(k)})^t w_{\text{train}}^{(k)}.\end{aligned}$$

For any $\eta \in [0, 1/L]$, we have $\left\|\widehat{w}_{\tau,\eta}^{(k)}\right\| \leqslant \left\|w_{\text{train}}^{(k)}\right\| = \left\|(w_{\text{train}}^{(k)})^* + (X_{\text{train}}^{(k)})^\dagger \xi_{\text{train}}^{(k)}\right\| \leqslant 2\sqrt{L}\sigma$ for any τ. Therefore, $\left\|w_{t,\eta}^{(k)}\right\|$ never exceeds the norm threshold and never gets truncated.

Noticing that $\Delta_{TbT}(\eta, P_k) = \frac{1}{2}(w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)})^\top H_{\text{train}}^{(k)}(w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)})$, we have

$$\Delta_{TbT}(\eta, P_k) = \frac{1}{2}(w_{\text{train}}^{(k)})^\top H_{\text{train}}^{(k)}(I - \eta H_{\text{train}}^{(k)})^{2t} w_{\text{train}}^{(k)}.$$

Taking the derivative of $\Delta_{TbT}(\eta, P_k)$ in η, we have

$$\frac{\partial}{\partial \eta}\Delta_{TbT}(\eta, P_k) = -t(w_{\text{train}}^{(k)})^\top (H_{\text{train}}^{(k)})^2 (I - \eta H_{\text{train}}^{(k)})^{2t-1} w_{\text{train}}^{(k)}.$$

Conditioning on $\mathcal{E}_k$, we know $1/L \leqslant \lambda_i(H_{\text{train}}^{(k)}) \leqslant L$ for all $i \in [n]$ and $H_{\text{train}}^{(k)}$ is full rank in the row span of $X_{\text{train}}^{(k)}$. Therefore, we know $\frac{\partial}{\partial \eta}\Delta_{TbT}(\eta, P_k) < 0$ for all $\eta \in [0, 1/L)$. Here, we assume $\left\|w_{\text{train}}^{(k)}\right\| > 0$, which happens with probability 1.

Overall, we know that conditioning on $\cap_{k\in[m]}\mathcal{E}_k$, every $\Delta_{TbT}(\eta, P_k)$ is strictly decreasing for $\eta \in [0, 1/L]$. Since $\widehat{F}_{TbT}(\eta) := \frac{1}{m}\sum_{k=1}^m \Delta_{TbT}(\eta, P_k)$, we know $\widehat{F}_{TbT}(\eta)$ is strictly decreasing when $\eta \in [0, 1/L]$.

At step size $\eta = 1/L$, we have

$$\begin{aligned}\Delta_{TbT}(\eta, P_k) =& \frac{1}{2}(w_{\text{train}}^{(k)})^\top H_{\text{train}}^{(k)}(I - \eta H_{\text{train}}^{(k)})^{2t} w_{\text{train}}^{(k)} \\ \leqslant& \frac{1}{2}L\left(1 - \frac{1}{L^2}\right)^t \left\|w_{\text{train}}^{(k)}\right\|^2 \leqslant 2L^2\sigma^2\left(1 - \frac{1}{L^2}\right)^t,\end{aligned}$$

where we upper bound $\left\|w_{\text{train}}^{(k)}\right\|^2$ by $4L\sigma^2$ at the last step. Therefore, we have $\widehat{F}_{TbT}(1/L) \leqslant 2L^2\sigma^2(1-\frac{1}{L^2})^t$. $\square$

Lower Bounding $\widehat{F}_{TbT}$ for $\eta \in (\widetilde{\eta}, \infty)$

In this section, we prove that the empirical meta objective is lower bounded by $\Omega(\sigma^2)$ with high probability for $\eta \in (\widetilde{\eta}, \infty)$. Step size $\widetilde{\eta}$ is defined such that there is a descent probability of diverging for any step size larger than $\widetilde{\eta}$. Then, we show the contribution from these truncated sequence will be enough to provide an $\Omega(\sigma^2)$ lower bound for $\widehat{F}_{TbT}$. The proof of Lemma 5.4.4 is given at the end of this section.

Lemma 5.4.4. *With probability at least* $1 - \exp(-\Omega(m))$,

$$\widehat{F}_{TbT}(\eta) \geqslant \frac{\sigma^2}{10L^8}$$

for all $\eta > \widetilde{\eta}$.

We define $\widetilde{\eta}$ as the smallest step size such that the contribution from the truncated sequence in the population meta objective exceeds certain threshold. The precise definition is as follows.

Definition D.2.5. Given a training task P, let $\mathcal{E}_1$ be the event that $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\text{train}}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{\text{train}}\| \leqslant \sqrt{d}\sigma$. Let $\bar{\mathcal{E}}_2(\eta)$ be the event that the GD sequence is truncated with step size η. Define $\widetilde{\eta}$ as follows,

$$\widetilde{\eta} = \inf\left\{\eta \geqslant 0 \middle| \mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\} \geqslant \frac{\sigma^2}{L^6}\right\}.$$

In the next lemma, we prove that for any fixed training set, $\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta')\right\} \geqslant$

$\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\}$ for any $\eta' \geqslant \eta$. This immediately implies that $\Pr\left[\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right]$ and $\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\}$ is non-decreasing in η.

Basically we need to show, conditioning on $\mathcal{E}_1$, if a GD sequence gets truncated at step size η, it must be also truncated for larger step sizes. Let $\{w'_{\tau,\eta}\}$ be the GD sequence without truncation. We only need to show that for any τ, if $\left\|w'_{\tau,\eta}\right\|$ exceeds the norm threshold, $\left\|w'_{\tau,\eta'}\right\|$ must also exceed the norm threshold for any $\eta' \geqslant \eta$. This is easy to prove if τ is odd because in this case $\left\|w'_{\tau,\eta}\right\|$ is always non-decreasing in η. The case when τ is even is trickier because there indeed exists certain range of η such that $\left\|w'_{\tau,\eta}\right\|$ is decreasing in η. We manage to prove that this problematic case cannot happen when $\left\|w'_{\tau,\eta}\right\|$ is at least $4\sqrt{L}\sigma$. The full proof of Lemma D.2.6 is deferred into Section D.2.2.

Lemma D.2.6. *Fixing a task P, let $\mathcal{E}_1$ and $\bar{\mathcal{E}}_2(\eta)$ be as defined in Definition D.2.5. We have*

$$\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta')\right\} \geqslant \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\},$$

for any $\eta' \geqslant \eta$.

In the next Lemma, we prove that $\widetilde{\eta}$ must lie within $(1/L, 3L)$. We prove this by showing that the GD sequence never gets truncated for $\eta \in [0, 2/L]$ and almost always gets truncated for $\eta \in [2.5L, \infty)$. The proof is deferred into Section D.2.2.

Lemma D.2.7. *Let $\widetilde{\eta}$ be as defined in Definition D.2.5. Suppose σ is a large constant c_1. Assume $t \geqslant c_2, d \geqslant c_4$ for some constants c_2, c_4. We have*

$$1/L < \widetilde{\eta} < 3L.$$

Now, we are ready to give the proof of Lemma 5.4.4.

Proof of Lemma 5.4.4. Let $\mathcal{E}_1$ and $\bar{\mathcal{E}}_2(\eta)$ be as defined in Definition D.2.5. For

the simplicity of the proof, we assume $\mathbb{E}\frac{1}{2}\|w_{t,\widetilde{\eta}} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta})\right\} \geqslant \frac{\sigma^2}{L^6}$. We will discuss the proof for the other case at the end, which is very similar.

Conditioning on $\mathcal{E}_1$, we know $\frac{1}{2}\|w_{t,\widetilde{\eta}} - w_{\text{train}}\|^2_{H_{\text{train}}} \leqslant 18L^2\sigma^2$. Therefore, we know $\Pr\big[\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta})\big] \geqslant \frac{1}{18L^8}$. For each task P_k, define $\mathcal{E}_1^{(k)}$ and $\bar{\mathcal{E}}_2^{(k)}(\eta)$ as the corresponding events on training set $S^{(k)}_{\text{train}}$. By Hoeffding's inequality, we know with probability at least $1 - \exp(-\Omega(m))$,

$$\frac{1}{m}\sum_{k=1}^{m} \mathbb{1}\left\{\mathcal{E}_1^{(k)} \cap \bar{\mathcal{E}}_2^{(k)}(\widetilde{\eta})\right\} \geqslant \frac{1}{20L^8}.$$

By Lemma D.2.6, we know $\mathbb{1}\left\{\mathcal{E}_1^{(k)} \cap \bar{\mathcal{E}}_2^{(k)}(\eta)\right\} \geqslant \mathbb{1}\left\{\mathcal{E}_1^{(k)} \cap \bar{\mathcal{E}}_2^{(k)}(\widetilde{\eta})\right\}$ for any $\eta \geqslant \widetilde{\eta}$. Then, we can lower bound $\widehat{F}_{TbT}$ for any $\eta > \widetilde{\eta}$ as follows,

$$\begin{aligned}
\widehat{F}_{TbT}(\eta) = \frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2_{H^{(k)}_{\text{train}}} \geqslant & \frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2_{H^{(k)}_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1^{(k)} \cap \bar{\mathcal{E}}_2^{(k)}(\eta)\right\} \\
\geqslant & 2\sigma^2\frac{1}{m}\sum_{k=1}^{m}\mathbb{1}\left\{\mathcal{E}_1^{(k)} \cap \bar{\mathcal{E}}_2^{(k)}(\eta)\right\} \\
\geqslant & 2\sigma^2\frac{1}{m}\sum_{k=1}^{m}\mathbb{1}\left\{\mathcal{E}_1^{(k)} \cap \bar{\mathcal{E}}_2^{(k)}(\widetilde{\eta})\right\} \geqslant \frac{\sigma^2}{10L^8},
\end{aligned}$$

where the second inequality lower bounds the loss for one task by $2\sigma^2$ when the sequence gets truncated.

We have assumed $\mathbb{E}\frac{1}{2}\|w_{t,\widetilde{\eta}} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta})\right\} \geqslant \frac{\sigma^2}{L^6}$ in the proof. Now, we show the proof also works when $\mathbb{E}\frac{1}{2}\|w_{t,\widetilde{\eta}} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta})\right\} < \frac{\sigma^2}{L^6}$ with slight changes. According to the definition and Lemma D.2.6, we are able to know that $\mathbb{E}\frac{1}{2}\|w_{t,\widetilde{\eta}} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\} > \frac{\sigma^2}{L^6}$ for all $\eta > \widetilde{\eta}$. At each training set S_{train}, we can define $\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta}')\right\}$ as $\lim_{\eta\to\widetilde{\eta}^+}\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\}$. Note that we also have $\Pr\big[\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta}')\big] \geqslant \frac{1}{18L^8}$. The remaining proof is the same as before as we substitute

$\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta})\right\}$ by $\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta}')\right\}$. □

Generalization for $\eta \in [1/L, \widetilde{\eta}]$

In this section, we show empirical meta objective $\widehat{F}_{TbT}$ is point-wise close to population meta objective F_{TbT} for all $\eta \in [1/L, \widetilde{\eta}]$.

Lemma D.2.4. *Suppose σ is a large constant c_1. Assume $t \geqslant c_2, d \geqslant c_4$ for certain constants c_2, c_4. With probability at least $1 - m\exp(-\Omega(d)) - O(t+m)\exp(-\Omega(m))$,*

$$|F_{TbT}(\eta) - \widehat{F}_{TbT}(\eta)| \leqslant \frac{\sigma^2}{L^3},$$

for all $\eta \in [1/L, \widetilde{\eta}]$,

In this section, we first show $\widehat{F}_{TbT}$ concentrates on F_{TbT} for any fixed η and then construct ϵ-net for $\widehat{F}_{TbT}$ and F_{TbT} for $\eta \in [1/L, \widetilde{\eta}]$. We give the proof of Lemma D.2.4 at the end.

We first show that for a fixed η, $\widehat{F}_{TbT}(\eta)$ is close to $F_{TbT}(\eta)$ with high probability. We prove the meta-loss on each task $\Delta_{TbT}(\eta, P_k)$ is $O(1)$-subexponential. Then we apply Bernstein's inequality to get the result. The proof is deferred into Section D.2.2. We will assume σ is a large constant and $t \geqslant c_2, d \geqslant c_4$ for some constants c_2, c_4 so that Lemma D.2.7 holds and $\widetilde{\eta}$ is a constant.

Lemma D.2.8. *Suppose σ is a constant. For any fixed η and any $1 > \epsilon > 0$, with probability at least $1 - \exp(-\Omega(\epsilon^2 m))$,*

$$\left|\widehat{F}_{TbT}(\eta) - F_{TbT}(\eta)\right| \leqslant \epsilon.$$

Next, we construct an ϵ-net for F_{TbT}. By the definition of $\widetilde{\eta}$, we know for any $\eta \leqslant \widetilde{\eta}$, the contribution from truncated sequences in $F_{TbT}(\eta)$ is small. We can show

the contribution from the un-truncated sequences is $O(t)$-lipschitz.

Lemma D.2.9. *Suppose σ is a large constant c_1. Assume $t \geqslant c_2, d \geqslant c_4$ for some constant c_2, c_4. There exists an $\frac{11\sigma^2}{L^4}$-net $N \subset [1/L, \widetilde{\eta}]$ for F_{TbT} with $|N| = O(t)$. That means, for any $\eta \in [1/L, \widetilde{\eta}]$,*

$$|F_{TbT}(\eta) - F_{TbT}(\eta')| \leqslant \frac{11\sigma^2}{L^4},$$

for $\eta' = \arg\min_{\eta'' \in N, \eta'' \leqslant \eta}(\eta - \eta'')$.

Proof of Lemma D.2.9. Let $\mathcal{E}_1$ and $\bar{\mathcal{E}}_2(\eta)$ be as defined in Definition D.2.5. For the simplicity of the proof, we assume $\mathbb{E}\frac{1}{2}\|w_{t,\widetilde{\eta}} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta})\right\} \leqslant \frac{\sigma^2}{L^6}$. We will discuss the proof for the other case at the end, which is very similar.

We can divide $\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2$ as follows,

$$\begin{aligned}
&\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \\
=&\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_2(\widetilde{\eta})\right\} + \mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta})\right\} \\
&+ \mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\bar{\mathcal{E}}_1\right\}.
\end{aligned}$$

We will construct an ϵ-net for the first term and show the other two terms are small. Let's first consider the third term. Since $\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2$ is $O(1)$-subexponential and $\Pr\left[\bar{\mathcal{E}}_1\right] \leqslant \exp(-\Omega(d))$, we have $\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\bar{\mathcal{E}}_1\right\} = O(1)\exp(-\Omega(d))$. Choosing d to be at least certain constant, we could know that $\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\bar{\mathcal{E}}_1\right\} \leqslant \sigma^2/L^4$.

For the second term, since $\mathbb{E}\frac{1}{2}\|w_{t,\widetilde{\eta}} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta})\right\} \leqslant \frac{\sigma^2}{L^6}$ and $\frac{1}{2}\|w_{t,\widetilde{\eta}} - w_{\text{train}}\|_{H_{\text{train}}}^2 \geqslant 2\sigma^2$ when $w_{t,\widetilde{\eta}}$ diverges, we know $\Pr\left[\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta})\right] \leqslant \frac{1}{2L^6}$. Then,

we can upper bound the second term as follows,

$$\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta})\right\} \leqslant 18L^2\sigma^2\frac{1}{2L^6} = \frac{9\sigma^2}{L^4}$$

Next, we show the first term $\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_2(\widetilde{\eta})\right\}$ has desirable Lipschitz condition. According to Lemma D.2.6, we know $\mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_2(\eta)\right\} \geqslant \mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_2(\widetilde{\eta})\right\}$ for any $\eta \leqslant \widetilde{\eta}$. Therefore, conditioning on $\mathcal{E}_1 \cap \mathcal{E}_2(\widetilde{\eta})$, we know $w_{t,\eta}$ never gets truncated for any $\eta \leqslant \widetilde{\eta}$. This means $w_{t,\eta} = B_{t,\eta}w_{\text{train}}$ with $B_{t,\eta} = (I - (I - \eta H_{\text{train}})^t)$. We can compute the derivative of $\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2$ as follows,

$$\frac{\partial}{\partial\eta}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 = \left\langle tH_{\text{train}}(I - \eta H_{\text{train}})^{t-1}w_{\text{train}}\right\rangle H_{\text{train}}(w_{t,\eta} - w_{\text{train}}).$$

Since $\|w_{t,\eta}\| = \|(I - (I - \eta H_{\text{train}})^t)w_{\text{train}}\| \leqslant 4\sqrt{L}\sigma$ and $\|w_{\text{train}}\| \leqslant 2\sqrt{L}\sigma$, we have $\|(I - \eta H_{\text{train}})^t w_{\text{train}}\| \leqslant 6\sqrt{L}\sigma$. We can also bound the term $\|(I - \eta H_{\text{train}})^{t-1}w_{\text{train}}\|$ with $\|(I - \eta H_{\text{train}})^t w_{\text{train}}\| + \|w_{\text{train}}\|$ by bounding the expanding directions using $\|(I - \eta H_{\text{train}})^t w_{\text{train}}\|$ and bounding the shrinking directions using $\|w_{\text{train}}\|$. Therefore, we can bound the derivative as follows,

$$\left|\frac{\partial}{\partial\eta}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2\right| \leqslant tL \times 8\sqrt{L}\sigma \times 6L\sqrt{L}\sigma = 48L^3\sigma^2 t.$$

Suppose σ is a constant, we know $\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_2(\widetilde{\eta})\right\}$ is $O(t)$-lipschitz. Therefore, there exists an $\frac{\sigma^2}{L^4}$-net N for $\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_2(\widetilde{\eta})\right\}$ with size $O(t)$. That means, for any $\eta \in [1/L, \widetilde{\eta}]$,

$$\left|\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_2(\widetilde{\eta})\right\} - \mathbb{E}\frac{1}{2}\|w_{t,\eta'} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_2(\widetilde{\eta})\right\}\right| \leqslant \frac{\sigma^2}{L^4}$$

for $\eta' = \arg\min_{\eta'' \in N, \eta'' \leqslant \eta}(\eta - \eta'')$. Note we construct the ϵ-net in a particular way such that η' is chosen as the largest step size in N that is at most η.

Combing with the upper bounds on the second term and the third term, we have for any $\eta \in [1/L, \widetilde{\eta}]$,

$$|F_{TbT}(\eta) - F_{TbT}(\eta')| \leqslant \frac{11\sigma^2}{L^4}$$

for $\eta' = \arg\min_{\eta'' \in N, \eta'' \leqslant \eta}(\eta - \eta'')$.

In the above analysis, we have assumed $\mathbb{E}\frac{1}{2}\|w_{t,\widetilde{\eta}} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta})\right\} \leqslant \frac{\sigma^2}{L^6}$. The proof can be easily generalized to the other case. We can define $\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta}')\right\}$ as $\lim_{\eta \to \widetilde{\eta}^-} \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\}$. Then the proof works as long as we substitute $\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta})\right\}$ by $\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widetilde{\eta}')\right\}$. We will also add $\widetilde{\eta}$ into the ϵ-net. □

In order to prove F_{TbT} is close to $\widehat{F}_{TbT}$ point-wise in $[1/L, \widetilde{\eta}]$, we still need to construct an ϵ-net for the empirical meta objective $\widehat{F}_{TbT}$.

Lemma D.2.10. *Suppose σ is a large constant c_1. Assume $t \geqslant c_2, d \geqslant c_4$ for certain constants c_2, c_4. With probability at least $1 - m\exp(-\Omega(d))$, there exists an $\frac{\sigma^2}{L^4}$-net $N' \subset [1/L, \widetilde{\eta}]$ for $\widehat{F}_{TbT}$ with $|N| = O(t + m)$. That means, for any $\eta \in [1/L, \widetilde{\eta}]$,*

$$|\widehat{F}_{TbT}(\eta) - \widehat{F}_{TbT}(\eta')| \leqslant \frac{\sigma^2}{L^4},$$

for $\eta' = \arg\min_{\eta'' \in N', \eta'' \leqslant \eta}(\eta - \eta'')$.

Proof of Lemma D.2.10. For each $k \in [m]$, let $\mathcal{E}_{1,k}$ be the event that $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X^{(k)}_{\text{train}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H^{(k)}_{\text{train}}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \left\|\xi^{(k)}_{\text{train}}\right\| \leqslant \sqrt{d}\sigma$. According to Lemma D.2.3 and Lemma D.5.1, we know with probability at least $1 - m\exp(-\Omega(d))$, $\mathcal{E}_{1,k}$'s hold for all $k \in [m]$. From now on, we assume all these events hold.

Recall that the empirical meta objective as follows,

$$\widehat{F}_{TbT}(\eta) := \frac{1}{m}\sum_{k=1}^{m}\Delta_{TbT}(\eta, P_k).$$

For any $k \in [m]$, let $\eta_{c,k}$ be the smallest step size such that $w_{t,\eta}^{(k)}$ gets truncated. If $\eta_{c,k} > \widehat{\eta}$, by similar argument as in Lemma D.2.9, we know $\Delta_{TbT}(\eta, P_k)$ is $O(t)$-Lipschitz in $[1/L, \widehat{\eta}]$ as long as σ is a constant. If $\eta_{c,k} \leqslant \widehat{\eta}$, by Lemma D.2.6 we know $w_{t,\eta}^{(k)}$ gets truncated for any $\eta \geqslant \eta_{c,k}$. This then implies that $\Delta_{TbT}(\eta, P_k)$ is a constant function for $\eta \in [\eta_{c,k}, \widehat{\eta}]$. We can also show that $\Delta_{TbT}(\eta, P_k)$ is $O(t)$-Lipschitz in $[1/L, \eta_{c,k})$. There might be a discontinuity in function value at $\eta_{c,k}$, so we need to add $\eta_{c,k}$ into the ϵ-net.

Overall, we know there exists an $\frac{\sigma^2}{L^4}$-net N' with $|N'| = O(t+m)$ for $\widehat{F}_{TbT}$. That means, for any $\eta \in [1/L, \widetilde{\eta}]$,

$$\left|\widehat{F}_{TbT}(\eta) - \widehat{F}_{TbT}(\eta')\right| \leqslant \frac{\sigma^2}{L^4}$$

for $\eta' = \arg\min_{\eta'' \in N', \eta'' \leqslant \eta}(\eta - \eta'')$. □

Finally, we combine Lemma D.2.8, Lemma D.2.9 and Lemma D.2.10 to prove that $\widehat{F}_{TbT}$ is point-wise close to F_{TbT} for $\eta \in [1/L, \widetilde{\eta}]$.

Proof of Lemma D.2.4. We assume σ as a constant in this proof. By Lemma D.2.8, we know with probability at least $1 - \exp(-\Omega(\epsilon^2 m))$, $\left|\widehat{F}_{TbT}(\eta) - F_{TbT}(\eta)\right| \leqslant \epsilon$ for any fixed η. By Lemma D.2.9, we know there exists an $\frac{11\sigma^2}{L^4}$-net N for F_{TbT} with size $O(t)$. By Lemma D.2.10, we know with probability at least $1 - m\exp(-\Omega(d))$, there exists an $\frac{\sigma^2}{L^4}$-net N' for $\widehat{F}_{TbT}$ with size $O(t+m)$. According to the proofs of Lemma D.2.9 and Lemma D.2.10, it's not hard to verify that $N \cup N'$ is still an $\frac{11\sigma^2}{L^4}$-net for $\widehat{F}_{TbT}$

and F_{TbT}. That means, for any $\eta \in [1/L, \widetilde{\eta}]$, we have

$$|F_{TbT}(\eta) - F_{TbT}(\eta')|, |\widehat{F}_{TbT}(\eta) - \widehat{F}_{TbT}(\eta')| \leqslant \frac{11\sigma^2}{L^4},$$

for $\eta' = \arg\min_{\eta'' \in N \cup N', \eta'' \leqslant \eta}(\eta - \eta'')$.

Taking a union bound over $N \cup N'$, we have with probability at least $1 - O(t + m)\exp(-\Omega(m))$,

$$\left|\widehat{F}_{TbT}(\eta) - F_{TbT}(\eta)\right| \leqslant \frac{\sigma^2}{L^4}$$

for all $\eta \in N \cup N'$.

Overall, we know with probability at least $1 - m\exp(-\Omega(d)) - O(t+m)\exp(-\Omega(m))$, for all $\eta \in [1/L, \widetilde{\eta}]$,

$$\begin{aligned}
&|F_{TbT}(\eta) - \widehat{F}_{TbT}(\eta)| \\
\leqslant &|F_{TbT}(\eta) - F_{TbT}(\eta')| + |\widehat{F}_{TbT}(\eta) - \widehat{F}_{TbT}(\eta')| + |\widehat{F}_{TbT}(\eta') - F_{TbT}(\eta')| \\
\leqslant &\frac{23\sigma^2}{L^4} \leqslant \frac{\sigma^2}{L^3},
\end{aligned}$$

where $\eta' = \arg\min_{\eta'' \in N \cup N', \eta'' \leqslant \eta}(\eta - \eta'')$. We use the fact that $L = 100$ in the last inequality. □

Proofs of Technical Lemmas

Proof of Lemma D.2.3. Recall that X_{train} is an $n \times d$ matix with $n = cd$ where $c \in [1/4, 3/4]$. According to Lemma D.5.4, with probability at least $1 - 2\exp(-t^2/2)$, we have

$$\sqrt{d} - \sqrt{cd} - t \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{d} + \sqrt{cd} + t,$$

for all $i \in [n]$.

Since $H_{\text{train}} = 1/nX_{\text{train}}^\top X_{\text{train}}$, we know $\lambda_i(H_{\text{train}}) = 1/n\sigma_i^2(X_{\text{train}})$. Since $c \in$

$[\frac{1}{4}, \frac{3}{4}]$, we have $\frac{1}{cd}(\sqrt{d}+\sqrt{cd})^2 \leqslant 100 - c'$ and $\frac{1}{cd}(\sqrt{d}-\sqrt{cd})^2 \geqslant \frac{1}{100} + c'$, for some constant c'. Therefore, we know with probability at least $1 - \exp(-\Omega(d))$,

$$\frac{1}{100} \leqslant \lambda_i(H_{\text{train}}) \leqslant 100,$$

for all $i \in [n]$.

Similarly, since there exists constant c'' such that $\sqrt{d}+\sqrt{cd} \leqslant (10 - c'')\sqrt{d}$ and $\sqrt{d}-\sqrt{cd} \geqslant (1/10 + c'')\sqrt{d}$, we know with probability at least $1 - \exp(-\Omega(d))$,

$$\frac{1}{10}\sqrt{d} \leqslant \sigma_i(X_{\text{train}}) \leqslant 10\sqrt{d},$$

for all $i \in [n]$. Choosing $L = 100$ finishes the proof. □

Proof of Lemma D.2.6. We prove that for any training set S_{train}, $\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta')\right\} \geqslant \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta')\right\}$ for any $\eta' > \eta$. This is trivially true if $\mathcal{E}_1$ is false on S_{train}. Therefore, we focus on the case when $\mathcal{E}_1$ holds for S_{train}. Suppose η_c is the smallest step size such that the GD sequence gets truncated. Let $\{w'_{\tau,\eta_c}\}$ be the GD sequence without truncation. There must exists $\tau \leqslant t$ such that $\left\|w'_{\tau,\eta_c}\right\| \geqslant 4\sqrt{L}\sigma$. We only need to prove that $\left\|w'_{\tau,\eta}\right\| \geqslant 4\sqrt{L}\sigma$ for any $\eta \geqslant \eta_c$. We prove this by showing the derivative of $\left\|w'_{\tau,\eta}\right\|^2$ in η is non-negative assuming $\left\|w'_{\tau,\eta}\right\|^2 \geqslant 4\sqrt{L}\sigma$.

Recall the recursion of $w'_{\tau,\eta}$ as $w'_{\tau,\eta} = w_{\text{train}} - (I - \eta H_{\text{train}})^\tau w_{\text{train}}$. If τ is an odd number, it's clear that $\frac{\partial}{\partial\eta}\left\|w'_{\tau,\eta}\right\|^2$ is non-negative at any $\eta \geqslant 0$. From now on, we assume τ is an even number. Actually in this case, $\frac{\partial}{\partial\eta}\left\|w'_{\tau,\eta}\right\|^2$ can be negative for some η. However, we can prove the derivative must be non-negative assuming $\left\|w'_{\tau,\eta}\right\|^2 \geqslant 4\sqrt{L}\sigma$.

Suppose the eigenvalue decomposition of H_{train} is $\sum_{i=1}^n \lambda_i u_i u_i^\top$ with $\lambda_1 \geqslant \cdots \lambda_n$. Denote c_i as $\langle w_{\text{train}} \rangle u_i$. Let λ_j be the smallest eigenvalue such that $(1 - \eta\lambda_j) \leqslant -1$.

This implies $\lambda_i \leqslant 2/\eta$ for any $i \geqslant j+1$. We can write down $\left\|w'_{\tau,\eta}\right\|^2$ as follows

$$\begin{aligned}\left\|w'_{\tau,\eta}\right\|^2 =& \sum_{i=1}^{j}\left(1-(1-\eta\lambda_i)^t\right)^2 c_i^2 + \sum_{i=j+1}^{n}\left(1-(1-\eta\lambda_i)^t\right)^2 c_i^2 \\ \leqslant& \sum_{i=1}^{j}\left(1-(1-\eta\lambda_i)^t\right)^2 c_i^2 + \|w_{\text{train}}\|^2.\end{aligned}$$

Since $\mathcal{E}_1$ holds, we know $\|w_{\text{train}}\|^2 \leqslant 4L\sigma^2$. Combining with $\left\|w'_{\tau,\eta}\right\|^2 \geqslant 16L\sigma^2$, we have $\sum_{i=1}^{j}\left(1-(1-\eta\lambda_i)^t\right)^2 c_i^2 \geqslant 12L\sigma^2$. We can lower bound the derivative as follows,

$$\begin{aligned}\frac{\partial}{\partial\eta}\|w_{\tau,\eta}\|^2 =& \sum_{i=1}^{j} 2t\lambda_i(1-\eta\lambda_i)^{t-1}\left(1-(1-\eta\lambda_i)^t\right)c_i^2 \\ &+ \sum_{i=j+1}^{n} 2t\lambda_i(1-\eta\lambda_i)^{t-1}\left(1-(1-\eta\lambda_i)^t\right)c_i^2 \\ \geqslant& 2t\sum_{i=1}^{j}\lambda_i(1-\eta\lambda_i)^{t-1}\left(1-(1-\eta\lambda_i)^t\right)c_i^2 - 2t\frac{2}{\eta}\sum_{i=j+1}^{n} c_i^2 \\ \geqslant& 2t\sum_{i=1}^{j}\lambda_i(1-\eta\lambda_i)^{t-1}\left(1-(1-\eta\lambda_i)^t\right)c_i^2 - 2t\times 8L\sigma^2/\eta.\end{aligned}$$

Then, we only need to show that $\sum_{i=1}^{j}\lambda_i(1-\eta\lambda_i)^{t-1}\left(1-(1-\eta\lambda_i)^t\right)c_i^2$ is larger than

$8L\sigma^2/\eta$. We have

$$\begin{aligned}
&\sum_{i=1}^{j} \lambda_i (1-\eta\lambda_i)^{t-1} \left(1-(1-\eta\lambda_i)^t\right) c_i^2 \\
=&\sum_{i=1}^{j} \lambda_i \frac{(1-\eta\lambda_i)^{t-1}}{1-(1-\eta\lambda_i)^t} \left(1-(1-\eta\lambda_i)^t\right)^2 c_i^2 \\
=&\sum_{i=1}^{j} \lambda_i \frac{(\eta\lambda_i-1)^{t-1}}{(\eta\lambda_i-1)^t-1} \left(1-(1-\eta\lambda_i)^t\right)^2 c_i^2 \\
=&\sum_{i=1}^{j} \lambda_i \frac{(\eta\lambda_i-1)^{t}}{(\eta\lambda_i-1)^t-1} \frac{1}{\eta\lambda_i-1} \left(1-(1-\eta\lambda_i)^t\right)^2 c_i^2 \\
\geqslant&\sum_{i=1}^{j} \frac{1}{\eta} \left(1-(1-\eta\lambda_i)^t\right)^2 c_i^2 \geqslant 12L\sigma^2/\eta > 8L\sigma^2/\eta.
\end{aligned}$$

□

Proof of Lemma D.2.7. Similar as the analysis in Lemma 5.4.3, conditioning on $\mathcal{E}_1$, we know the GD sequence never exceeds the norm threshold for any $\eta \in [0, 2/L]$. This then implies

$$\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\} = 0,$$

for all $\eta \in [0, 2/L]$.

Let $\{w'_{\tau,\eta}\}$ be the GD sequence without truncation. For any step size $\eta \in [2.5L, \infty]$, conditioning on $\mathcal{E}_1$, we have

$$\left\|w'_{t,\eta}\right\| \geqslant \left((\eta/L-1)^t - 1\right)\|w_{\text{train}}\| \geqslant \left(1.5^t - 1\right)\left(\frac{\sigma}{4\sqrt{L}} - 1\right) \geqslant 4\sqrt{L}\sigma,$$

where the last inequality holds as long as $\sigma \geqslant 5\sqrt{L}, t \geqslant c_2$ for some constant c_2. Therefore, we know when $\eta \in [2.5L, \infty)$, $\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\} = \mathbb{1}\left\{\mathcal{E}_1\right\}$. Then, we have for

any $\eta \geqslant 2.5L$,

$$\begin{aligned}
&\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\} \\
\geqslant &\frac{1}{2L}\left(4\sqrt{L}\sigma - 2\sqrt{L}\sigma\right)^2 \Pr[\mathcal{E}_1] \geqslant 2\sigma^2 \Pr[\mathcal{E}_1] \geqslant \frac{\sigma^2}{L^3},
\end{aligned}$$

where the last inequality uses $\Pr[\mathcal{E}_1] \geqslant 1 - \exp(-\Omega(d))$ and assume $d \geqslant c_4$ for some constant c_4.

Overall, we know $\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\}$ equals zero for all $\eta \in [0, 2/L]$ and is at least $\frac{\sigma^2}{L^3}$ for all $\eta \in [2.5L, \infty)$. By definition, we know $\widetilde{\eta} \in (1/L, 3L)$.

□

Proof of Lemma D.2.8. Recall that $\widehat{F}_{TbT}(\eta) := \frac{1}{m}\sum_{k=1}^m \Delta_{TbT}(\eta, P_k)$. We prove that each $\Delta_{TbT}(\eta, P_k)$ is $O(1)$-subexponential. We can further write $\Delta_{TbT}(\eta, P_k)$ as follows,

$$\begin{aligned}
\Delta_{TbT}(\eta, P_k) =&\frac{1}{2}\left\|w_{t,\eta}^{(k)} - w_k^* - (X_{\text{train}}^{(k)})^\dagger \xi_{\text{train}}^{(k)}\right\|_{H_{\text{train}}^{(k)}}^2 \\
\leqslant&\frac{1}{2}\left\|w_{t,\eta}^{(k)} - w_k^*\right\|^2 \left\|H_{\text{train}}^{(k)}\right\| + \frac{1}{2n}\left\|\xi_{\text{train}}^{(k)}\right\|^2 \\
&+ \left\|w_{t,\eta}^{(k)} - w_k^*\right\| \left(\frac{1}{\sqrt{n}}\left\|\xi_{\text{train}}^{(k)}\right\|\right)\left(\frac{1}{\sqrt{n}}\left\|X_{\text{train}}^{(k)}\right\|\right).
\end{aligned}$$

We can write $\left\|H_{\text{train}}^{(k)}\right\|$ as $\sigma_{\max}^2(\frac{1}{\sqrt{n}}X_{\text{train}}^{(k)})$. According to Lemma D.5.3, we know that $\sigma_{\max}(X_{\text{train}}^{(k)}) - \mathbb{E}\sigma_{\max}(X_{\text{train}}^{(k)})$ is $O(1)$-subgaussian, which implies that $\sigma_{\max}(\frac{1}{\sqrt{n}}X_{\text{train}}^{(k)}) - \mathbb{E}\sigma_{\max}(\frac{1}{\sqrt{n}}X_{\text{train}}^{(k)})$ is $O(1/\sqrt{d})$-subgaussian. Since $\mathbb{E}\sigma_{\max}(\frac{1}{\sqrt{n}}X_{\text{train}}^{(k)})$ is a constant, we know $\sigma_{\max}(\frac{1}{\sqrt{n}}X_{\text{train}}^{(k)})$ is $O(1)$-subgaussian and $\sigma_{\max}^2(\frac{1}{\sqrt{n}}X_{\text{train}}^{(k)})$ is $O(1)$-subexponential. Similarly, we know both the terms $\frac{1}{2n}\left\|\xi_{\text{train}}^{(k)}\right\|^2$ and $\left(\frac{1}{\sqrt{n}}\left\|X_{\text{train}}^{(k)}\right\|\right)\left(\frac{1}{\sqrt{n}}\left\|\xi_{\text{train}}^{(k)}\right\|\right)$ are $O(1)$-subexponential.

Suppose σ is a constant, we know $\left\|w_{t,\eta}^{(k)} - w_k^*\right\|$ is upper bounded by a constant. Then, we know $\Delta_{TbT}(\eta, P_k)$ is $O(1)$-subexponential. Therefore, $\widehat{F}_{TbT}(\eta)$ is the average of m i.i.d. $O(1)$-subexponential random variables. By standard concentration inequality, we know for any $1 > \epsilon > 0$, with probability at least $1 - \exp(-\Omega(\epsilon^2 m))$,

$$\left|\widehat{F}_{TbT}(\eta) - F_{TbT}(\eta)\right| \leqslant \epsilon.$$

□

D.2.3 Train-by-Validation (GD)

In this section, we show that the optimal step size under $\widehat{F}_{TbV}$ is $\Theta(1/t)$. Furthermore, we show under this optimal step size, GD sequence makes constant progress towards the ground truth. Precisely, we prove the following theorem.

Theorem D.2.2. *Let the meta objective $\widehat{F}_{TbV(n_1,n_2)}(\eta)$ be as defined in Equation 5.4 with $n_1, n_2 \in [d/4, 3d/4]$. Assume noise level σ is a large constant c_1. Assume unroll length $t \geqslant c_2$, number of training tasks $m \geqslant c_3$ and dimension $d \geqslant c_4 \log(t)$ for certain constants c_2, c_3, c_4. With probability at least 0.99 in the sampling of training tasks, we have*

$$\eta_{valid}^* = \Theta(1/t) \text{ and } \mathbb{E}\left\|w_{t,\eta_{valid}^*} - w^*\right\|^2 = \|w^*\|^2 - \Omega(1)$$

for all $\eta_{valid}^* \in \arg\min_{\eta \geqslant 0} \widehat{F}_{TbV(n_1,n_2)}(\eta)$, where the expectation is taken over new tasks.

In this section, we still use L to denote constant 100. We start from analyzing the behavior of the population meta-objective F_{TbV} for step sizes within $[0, 1/L]$. We show the optimal step size within this range is $\Theta(1/t)$ and GD sequence moves towards w^* under the optimal step size. The following lemma is a formal version

of Lemma 5.4.6. This serves as step 1 in Section D.2.1. We defer the proof of Lemma D.2.11 into Section D.2.3.

Lemma D.2.11. *Suppose noise level σ is a large enough constant c_1. Assume unroll length $t \geqslant c_2$ and dimension $d \geqslant c_4$ for some constants c_2, c_4. There exist $\eta_1, \eta_2, \eta_3 = \Theta(1/t)$ with $\eta_1 < \eta_2 < \eta_3$ such that*

$$F_{TbV}(\eta_2) \leqslant \frac{1}{2}\|w^*\|^2 - \frac{9}{10}C + \frac{\sigma^2}{2}$$

$$F_{TbV}(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{6}{10}C + \frac{\sigma^2}{2}, \forall \eta \in [0, \eta_1] \cup [\eta_3, 1/L]$$

where C is a positive constant.

To relate the behavior of F_{TbV} to the behavior of $\widehat{F}_{TbV}$, we prove the following generalization result for step sizes in $[0, 1/L]$. The following lemma is a formal version of Lemma 5.4.8. This serves as step 3 in Section D.2.1. The proof is deferred into Section D.2.3.

Lemma D.2.12. *For any $1 > \epsilon > 0$, assume $d \geqslant c_4 \log(1/\epsilon)$ for some constant c_4. With probability at least $1 - O(1/\epsilon)\exp(-\Omega(\epsilon^2 m))$,*

$$|\widehat{F}_{TbV}(\eta) - F_{TbV}(\eta)| \leqslant \epsilon,$$

for all $\eta \in [0, 1/L]$.

In Lemma D.2.13, we show the empirical meta objective $\widehat{F}_{TbV}$ is high for all step size larger than $1/L$, which then implies $\eta^*_{\text{valid}} \in [0, 1/L]$. The following lemma is a formal version of Lemma 5.4.7. This serves as step 2 in Section D.2.1. We prove this lemma in Section D.2.3.

Lemma D.2.13. *Suppose σ is a large constant. Assume $t \geqslant c_2, d \geqslant c_4 \log(t)$ for some constants c_2, c_4. With probability at least $1 - \exp(-\Omega(m))$,*

$$\widehat{F}_{TbV}(\eta) \geqslant C'\sigma^2 + \frac{1}{2}\sigma^2,$$

for all $\eta \geqslant 1/L$, where C' is a positive constant independent with σ.

Combining Lemma D.2.11, Lemma D.2.12 and Lemma D.2.13, we give the proof of Theorem D.2.2.

Proof of Theorem D.2.2. According to Lemma D.2.11, we know as long as d and t are larger than certain constants, there exists $\eta_1, \eta_2, \eta_3 = \Theta(1/t)$ with $\eta_1 < \eta_2 < \eta_3$ such that

$$F_{TbV}(\eta_2) \leqslant \frac{1}{2}\|w^*\|^2 - \frac{9}{10}C + \sigma^2/2$$
$$F_{TbV}(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{6}{10}C + \sigma^2/2, \forall \eta \in [0, \eta_1] \cup [\eta_3, 1/L],$$

for some positive constant C.

Choosing $\epsilon = \min(1, C/10)$ in Lemma D.2.12, we know as long as d is larger than certain constant, with probability at least $1 - \exp(-\Omega(m))$,

$$|\widehat{F}_{TbV}(\eta) - F_{TbV}(\eta)| \leqslant C/10,$$

for all $\eta \in [0, 1/L]$.

Therefore,

$$\widehat{F}_{TbV}(\eta_2) \leqslant \frac{1}{2}\|w^*\|^2 - \frac{8}{10}C + \sigma^2/2$$
$$\widehat{F}_{TbV}(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{7}{10}C + \sigma^2/2, \forall \eta \in [0, \eta_1] \cup [\eta_3, 1/L].$$

By Lemma D.2.13, we know as long as $t \geqslant c_2, d \geqslant c_4 \log(t)$ for some constants

c_2, c_4, with probability at least $1 - \exp(-\Omega(m))$,

$$\widehat{F}_{TbV}(\eta) \geqslant C'\sigma^2 + \frac{1}{2}\sigma^2,$$

for all $\eta \geqslant 1/L$. As long as $\sigma \geqslant 1/\sqrt{C'}$, we have $\widehat{F}_{TbV}(\eta) \geqslant 1 + \frac{1}{2}\sigma^2$ for all $\eta \geqslant 1/L$. Combining with $\widehat{F}_{TbV}(\eta_2) \leqslant \frac{1}{2}\|w^*\|^2 - \frac{8}{10}C + \sigma^2/2$, we know $\eta^*_{\text{valid}} \in [0, 1/L]$. Furthermore, since $\widehat{F}_{TbV}(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{7}{10}C + \sigma^2/2, \forall \eta \in [0, \eta_1] \cup [\eta_3, 1/L]$, we have $\eta_1 \leqslant \eta^*_{\text{valid}} \leqslant \eta_3$.

Recall that $\eta_1, \eta_3 = \Theta(1/t)$, we know $\eta^*_{\text{valid}} = \Theta(1/t)$. At the optimal step size, we have

$$F_{TbV}(\eta^*_{\text{valid}}) \leqslant \widehat{F}_{TbV}(\eta^*_{\text{valid}}) + C/10 \leqslant \widehat{F}_{TbV}(\eta_2) + C/10 \leqslant \frac{1}{2}\|w^*\|^2 - \frac{7}{10}C + \sigma^2/2.$$

Since $F_{TbV}(\eta^*_{\text{valid}}) = \mathbb{E}\frac{1}{2}\left\|w_{t,\eta^*_{\text{valid}}} - w^*\right\|^2 + \sigma^2/2$, we have

$$\mathbb{E}\left\|w_{t,\eta^*_{\text{valid}}} - w^*\right\|^2 \leqslant \|w^*\|^2 - \frac{7}{5}C.$$

Choosing m to be at least certain constant, this holds with probability at least 0.99.

□

Behavior of F_{TbV} for $\eta \in [0, 1/L]$

In this section, we study the behavior of F_{TbV} when $\eta \in [0, 1/L]$. We prove the following Lemma.

Lemma D.2.11. *Suppose noise level σ is a large enough constant c_1. Assume unroll length $t \geqslant c_2$ and dimension $d \geqslant c_4$ for some constants c_2, c_4. There exist $\eta_1, \eta_2, \eta_3 =$*

$\Theta(1/t)$ *with* $\eta_1 < \eta_2 < \eta_3$ *such that*

$$F_{TbV}(\eta_2) \leqslant \frac{1}{2}\|w^*\|^2 - \frac{9}{10}C + \frac{\sigma^2}{2}$$

$$F_{TbV}(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{6}{10}C + \frac{\sigma^2}{2}, \forall \eta \in [0, \eta_1] \cup [\eta_3, 1/L]$$

where C *is a positive constant.*

It's not hard to verify that $F_{TbV}(\eta) = \mathbb{E}1/2\|w_{t,\eta} - w^*\|^2 + \sigma^2/2$. For convenience, denote $Q(\eta) := 1/2\|w_{t,\eta} - w^*\|^2$. In order to prove Lemma D.2.11, we only need to show that $\mathbb{E}Q(\eta_2) \leqslant \frac{1}{2}\|w^*\|^2 - \frac{9}{10}C$ and $\mathbb{E}Q(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{6}{10}C$ for all $\eta \in [0, \eta_1] \cup [\eta_3, 1/L]$. In Lemma D.2.14, we first show that this happens with high probability over the sampling of tasks.

Lemma D.2.14. *Suppose noise level* σ *is a large enough constant* c_1*. Assume unroll length* $t \geqslant c_2$ *for certain constant* c_2*. Then, with probability at least* $1 - \exp(-\Omega(d))$ *over the sampling of tasks, there exists* $\eta_1, \eta_2, \eta_3 = \Theta(1/t)$ *with* $\eta_1 < \eta_2 < \eta_3$ *such that*

$$Q(\eta_2) := \frac{1}{2}\|w_{t,\eta_2} - w^*\|^2 \leqslant \frac{1}{2}\|w^*\|^2 - C$$

$$Q(\eta) := \frac{1}{2}\|w_{t,\eta} - w^*\|^2 \geqslant \frac{1}{2}\|w^*\|^2 - \frac{C}{2}, \forall \eta \in [0, \eta_1] \cup [\eta_3, 1/L]$$

where C *is a positive constant.*

Since we are in the small step size regime, we know the GD sequence converges with high probability and will not be truncated. For now, let's assume

$w_{t,\eta} = B_{t,\eta} w^*_{\text{train}} + B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}$, where $B_{t,\eta} = I - (I - \eta H_{\text{train}})^t$. We have

$$\begin{aligned}
Q(\eta) =& \frac{1}{2}\left\| B_{t,\eta} w^*_{\text{train}} + B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}} - w^* \right\|^2 \\
=& \frac{1}{2}\| B_{t,\eta} w^*_{\text{train}} - w^* \|^2 + \frac{1}{2}\left\| B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}} \right\|^2 \\
&+ \left\langle B_{t,\eta} w^*_{\text{train}} - w^* \right\rangle B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}} \\
=& \frac{1}{2}\| w^* \|^2 + \frac{1}{2}\| B_{t,\eta} w^*_{\text{train}} \|^2 + \frac{1}{2}\left\| B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}} \right\|^2 - \left\langle B_{t,\eta} w^*_{\text{train}} \right\rangle w^* \\
&+ \left\langle B_{t,\eta} w^*_{\text{train}} - w^* \right\rangle B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}.
\end{aligned}$$

In Lemma D.2.15, we are going to show that with high probability the crossing term $\left\langle B_{t,\eta} w^*_{\text{train}} - w^* \right\rangle B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}$ is negligible for all $\eta \in [0, 1/L]$. By Hoeffding's inequality, we know the crossing term is small for any fixed η. Constructing an ϵ-net for the crossing term in η, we can take a union bound and show it's small for all $\eta \in [0, 1/L]$. We defer the proof of Lemma D.2.15 to Section D.2.3.

Lemma D.2.15. *Assume σ is a constant. For any $1 > \epsilon > 0$, we know with probability at least $1 - O(1/\epsilon)\exp(-\Omega(\epsilon^2 d))$,*

$$\left| \left\langle B_{t,\eta} w^*_{train} - w^* \right\rangle B_{t,\eta}(X_{train})^\dagger \xi_{train} \right| \leqslant \epsilon,$$

for all $\eta \in [0, 1/L]$.

Denote

$$G(\eta) := \frac{1}{2}\| w^* \|^2 + \frac{1}{2}\| B_{t,\eta} w^*_{\text{train}} \|^2 + \frac{1}{2}\left\| B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}} \right\|^2 - \left\langle B_{t,\eta} w^*_{\text{train}} \right\rangle w^*.$$

Choosing $\epsilon = C/4$ in Lemma D.2.15, now we only need to show that $G(\eta_2) \leqslant \|w^*\|^2 - 5C/4$ and $G(\eta) \geqslant \|w^*\|^2 - C/4$ for all $\eta \in [0, \eta_1] \cup [\eta_3, 1/L]$.

We first show that there exists $\eta_2 = \Theta(1/t)$ such that $G(\eta_2) \leqslant \frac{1}{2}\|w^*\|^2 - 5C/4$ for

some constant C. It's not hard to show that $\frac{1}{2}\|B_{t,\eta}w^*_{\text{train}}\|^2 + \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2 = O(\eta^2 t^2 \sigma^2)$. In Lemma D.2.16, we show that the improvement $\langle B_{t,\eta}w^*_{\text{train}}\rangle w^* = \Omega(\eta t)$ is linear in η. Therefore there exists $\eta_2 = \Theta(1/t)$ such that $G(\eta_2) \leqslant \frac{1}{2}\|w^*\|^2 - 5C/4$ for some constant C. We defer the proof of Lemma D.2.16 to Section D.2.3.

Lemma D.2.16. *For any fixed $\eta \in [0, L/t]$ with probability at least $1 - \exp(-\Omega(d))$,*

$$\langle B_{t,\eta}w^*_{train}\rangle w^* \geqslant \frac{\eta t}{16L}.$$

To lower bound $G(\eta)$ for small η, we notice

$$G(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \langle B_{t,\eta}w^*_{\text{train}}\rangle w^*.$$

We can show that $\langle B_{t,\eta}w^*_{\text{train}}\rangle w^* = O(\eta t)$. Therefore, there exists $\eta_1 = \Theta(1/t)$ such that $\langle B_{t,\eta}w^*_{\text{train}}\rangle w^* \leqslant C/4$ for all $\eta \in [0, \eta_1]$.

To lower bound $G(\eta)$ for large η, we lower bound $G(\eta)$ using the noise square term,

$$G(\eta) \geqslant \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2.$$

We show that with high probability the term $\left\|B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2 = \Omega(\sigma^2)$ for all $\eta \in [\log(2)L/t, 1/L]$. Therefore, as long as σ is larger than some constant, there exists $\eta_3 = \Theta(1/t)$ such that $G(\eta) \geqslant \frac{1}{2}\|w^*\|^2$ for all $\eta \in [\eta_3, 1/L]$.

Combing Lemma D.2.15 and Lemma D.2.16, we can now give a complete proof for Lemma D.2.14.

Proof of Lemma D.2.14. Recall that

$$\begin{aligned}Q(\eta) =&\frac{1}{2}\|B_{t,\eta}w^*_{\text{train}} - w^*\|^2 + \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger\xi_{\text{train}}\right\|^2\\ &+\left\langle B_{t,\eta}w^*_{\text{train}} - w^*\right\rangle B_{t,\eta}(X_{\text{train}})^\dagger\xi_{\text{train}}\\ =&G(\eta) + \left\langle B_{t,\eta}w^*_{\text{train}} - w^*\right\rangle B_{t,\eta}(X_{\text{train}})^\dagger\xi_{\text{train}}\end{aligned}$$

We first show that with probability at least $1-\exp(-\Omega(d))$, there exist $\eta_1,\eta_2,\eta_3 = \Theta(1/t)$ with $\eta_1<\eta_2<\eta_3$ such that $G(\eta_2)\leqslant 1/2\|w^*\|^2 - 5C/4$ and $G(\eta)\geqslant 1/2\|w^*\|^2 - C/4$ for all $\eta\in[0,\eta_1]\cup[\eta_3,1/L]$.

According to Lemma D.2.3, we know with probability at least $1-\exp(-\Omega(d))$, $\sqrt{d}/\sqrt{L}\leqslant\sigma_i(X_{\text{train}})\leqslant\sqrt{Ld}$ and $1/L\leqslant\lambda_i(H_{\text{train}})\leqslant L$ for all $i\in[n]$ with $L=100$.

Upper bounding $G(\eta_2)$: We can expand $G(\eta)$ as follows:

$$\begin{aligned}G(\eta) :=&\frac{1}{2}\|B_{t,\eta}w^*_{\text{train}} - w^*\|^2 + \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger\xi_{\text{train}}\right\|^2\\ =&\frac{1}{2}\|w^*\|^2 + \frac{1}{2}\|B_{t,\eta}w^*_{\text{train}}\|^2 + \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger\xi_{\text{train}}\right\|^2 - \left\langle B_{t,\eta}w^*_{\text{train}}\right\rangle w^*.\end{aligned}$$

Recall that $B_{t,\eta} = I-(I-\eta H_{\text{train}})^t$, for any vector w in the span of H_{train},

$$\|B_{t,\eta}w\| = \left\|\left(I-(I-\eta H_{\text{train}})^t\right)w\right\|\leqslant L\eta t\,\|w\|\,.$$

According to Lemma D.5.1, we know with probability at least $1-\exp(-\Omega(d))$, $\|\xi_{\text{train}}\|\leqslant\sqrt{d}\sigma$. Therefore, we have

$$\frac{1}{2}\|B_{t,\eta}w^*_{\text{train}}\|^2 + \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger\xi_{\text{train}}\right\|^2\leqslant L^2\eta^2t^2/2 + L^3\eta^2t^2\sigma^2/2\leqslant L^3\eta^2t^2\sigma^2,$$

where the second inequality uses $\sigma, L\geqslant 1$. According to Lemma D.2.16, for any fixed $\eta\in[0,L/t]$, with probability at least $1-\exp(-\Omega(d))$, $\left\langle B_{t,\eta}w^*_{\text{train}}\right\rangle w^*\geqslant\frac{\eta t}{16L}$.

Therefore,

$$G(\eta) \leqslant \frac{1}{2}\|w^*\|^2 + L^3\eta^2t^2\sigma^2 - \frac{\eta t}{16L} \leqslant \frac{1}{2}\|w^*\|^2 - \frac{\eta t}{32L},$$

where the second inequality holds as long as $\eta \leqslant \frac{1}{32L^4\sigma^2 t}$. Choosing $\eta_2 := \frac{1}{32L^4\sigma^2 t}$, we have

$$G(\eta_2) \leqslant \frac{1}{2}\|w^*\|^2 - \frac{1}{1024L^5\sigma^2} = \frac{1}{2}\|w^*\|^2 - \frac{5C}{4},$$

where $C = \frac{1}{819.2L^5\sigma^2}$. Note C is a constant as σ, L are constants.

Lower bounding $G(\eta)$ for $\eta \in [0, \eta_1]$: Now, we prove that there exists $\eta_1 = \Theta(1/t)$ with $\eta_1 < \eta_2$ such that for any $\eta \in [0, \eta_1], G(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{C}{4}$. Recall that

$$\begin{aligned} G(\eta) =& \frac{1}{2}\|w^*\|^2 + \frac{1}{2}\|B_{t,\eta}w^*_{\text{train}}\|^2 + \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2 - \langle B_{t,\eta}w^*_{\text{train}}\rangle w^*. \\ \geqslant& \frac{1}{2}\|w^*\|^2 - \langle B_{t,\eta}w^*_{\text{train}}\rangle w^*. \end{aligned}$$

Since $|\langle B_{t,\eta}w^*_{\text{train}}\rangle w^*| \leqslant L\eta t$, we know for any $\eta \in [0, \eta_1]$,

$$G(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - L\eta_1 t.$$

Choosing $\eta_1 = \frac{C}{4Lt}$, we have for any $\eta \in [0, \eta_1]$,

$$G(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{C}{4}.$$

Lower bounding $G(\eta)$ for $\eta \in [\eta_3, 1/L]$: Now, we prove that there exists $\eta_3 = \Theta(1/t)$ with $\eta_3 > \eta_2$ such that for all $\eta \in [\eta_3, 1/L]$,

$$G(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{C}{4}.$$

Recall that

$$G(\eta) = \frac{1}{2}\|B_{t,\eta}w^*_{\text{train}} - w^*\|^2 + \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2 \geqslant \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2.$$

According to Lemma D.5.1, we know with probability at least $1-\exp(-\Omega(d))$, $\frac{\sqrt{d}\sigma}{2\sqrt{2}} \leqslant \|\xi_{\text{train}}\|$. Therefore,

$$\left\|B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2 \geqslant \left(1 - e^{-\eta t/L}\right)^2 \frac{\sigma^2}{8L} \geqslant \frac{\sigma^2}{32L},$$

where the last inequality assumes $\eta \geqslant \log(2)L/t$. As long as $t \geqslant \log(2)L^2$, we have $\log(2)L/t \leqslant 1/L$. Choosing $\eta_3 = \log(2)L/t$, we know for all $\eta \in [\eta_3, 1/L]$,

$$G(\eta) \geqslant \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2 \geqslant \frac{\sigma^2}{64L}.$$

Note that $\frac{1}{2}\|w^*\|^2 = 1/2$. Therefore, as long as $\sigma \geqslant 8\sqrt{L}$, we have

$$G(\eta) \geqslant \frac{1}{2}\|w^*\|^2$$

for all $\eta \in [\eta_3, 1/L]$.

Overall, we have shown that there exist $\eta_1, \eta_2, \eta_3 = \Theta(1/t)$ with $\eta_1 < \eta_2 < \eta_3$ such that $G(\eta_2) \leqslant 1/2\|w^*\|^2 - 5C/4$ and $G(\eta) \geqslant 1/2\|w^*\|^2 - C/4$ for all $\eta \in [0, \eta_1] \cup [\eta_3, 1/L]$. Recall that $Q(\eta) = G(\eta) + \langle B_{t,\eta}w^*_{\text{train}} - w^* \rangle B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}$. Choosing $\epsilon = C/4$ in Lemma D.2.15, we know with probability at least $1 - \exp(-\Omega(d))$, $\left|\langle B_{t,\eta}w^*_{\text{train}} - w^* \rangle B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right| \leqslant C/4$ for all $\eta \in [0, 1/L]$. Therefore, we know $Q(\eta_2) \leqslant 1/2\|w^*\|^2 - C$ and $Q(\eta) \geqslant 1/2\|w^*\|^2 - C/2$ for all $\eta \in [0, \eta_1] \cup [\eta_3, 1/L]$. □

Next, we give the proof of Lemma D.2.11.

Proof of Lemma D.2.11. Recall that $F_{TbV}(\eta) = \mathbb{E}1/2\|w_{t,\eta} - w^*\|^2 + \frac{\sigma^2}{2}$. For convenience, denote $Q(\eta) := 1/2\|w_{t,\eta} - w^*\|^2$. In order to prove Lemma D.2.11, we

only need to show that $\mathbb{E}Q(\eta_2) \leqslant \frac{1}{2}\|w^*\|^2 - \frac{9}{10}C$ and $\mathbb{E}Q(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{6}{10}C$ for all $\eta \in [0, \eta_1] \cup [\eta_3, 1/L]$.

According to Lemma D.2.14, as long as σ is a large enough constant c_1 and t is at least certain constant c_2, with probability at least $1 - \exp(-\Omega(d))$ over the sampling of S_{train}, there exists $\eta_1, \eta_2, \eta_3 = \Theta(1/t)$ with $\eta_1 < \eta_2 < \eta_3$ such that

$$Q(\eta_2) := 1/2\|w_{t,\eta_2} - w^*\|^2 \leqslant \frac{1}{2}\|w^*\|^2 - C$$

$$Q(\eta) := 1/2\|w_{t,\eta} - w^*\|^2 \geqslant \frac{1}{2}\|w^*\|^2 - \frac{C}{2}, \forall \eta \in [0, \eta_1] \cup [\eta_3, 1/L]$$

where C is a positive constant. Call this event $\mathcal{E}$. Suppose the probability that $\mathcal{E}$ happens is $1 - \delta$. We can write $\mathbb{E}Q(\eta)$ as follows,

$$\mathbb{E}Q(\eta) = \mathbb{E}[Q(\eta)|\mathcal{E}]\Pr[\mathcal{E}] + \mathbb{E}[Q(\eta)|\bar{\mathcal{E}}]\Pr[\bar{\mathcal{E}}].$$

According to the algorithm, we know $\|w_{t,\eta}\|$ is always bounded by $4\sqrt{L}\sigma$. Therefore, $Q(\eta) := 1/2\|w_{t,\eta} - w^*\|^2 \leqslant 13L\sigma^2$. When $\eta = \eta_2$, we have

$$\begin{aligned}
\mathbb{E}Q(\eta_2) \leqslant& \left(\frac{1}{2}\|w^*\|^2 - C\right)(1-\delta) + 13L\sigma^2\delta \\
=& \frac{1}{2}\|w^*\|^2 - \frac{\delta}{2} - C + (C + 13L\sigma^2)\delta \\
\leqslant& \frac{1}{2}\|w^*\|^2 - \frac{9C}{10},
\end{aligned}$$

where the last inequality assumes $\delta \leqslant \frac{C}{10C+130L\sigma^2}$.

When $\eta \in [0, \eta_1] \cup [\eta_3, 1/L]$, we have

$$\begin{aligned}
\mathbb{E}Q(\eta_2) \geqslant & \left(\frac{1}{2}\|w^*\|^2 - \frac{C}{2}\right)(1-\delta) - 13L\sigma^2\delta \\
= & \frac{1}{2}\|w^*\|^2 - \frac{\delta}{2} - (1-\delta)\frac{C}{2} - 13L\sigma^2\delta \\
\geqslant & \frac{1}{2}\|w^*\|^2 - \frac{C}{2} - (1/2 + 13L\sigma^2)\delta \\
\geqslant & \frac{1}{2}\|w^*\|^2 - \frac{6C}{10},
\end{aligned}$$

where the last inequality holds as long as $\delta \leqslant \frac{C}{5C+130L\sigma^2}$.

According to Lemma D.2.14, we know $\delta \leqslant \exp(-\Omega(d))$. Therefore, the conditions for δ can be satisfied as long as d is larger than certain constant. □

Generalization for $\eta \in [0, 1/L]$

In this section, we show $\widehat{F}_{TbV}$ is point-wise close to F_{TbV} for all $\eta \in [0, 1/L]$. Recall Lemma D.2.12 as follows.

Lemma D.2.12. *For any $1 > \epsilon > 0$, assume $d \geqslant c_4 \log(1/\epsilon)$ for some constant c_4. With probability at least $1 - O(1/\epsilon)\exp(-\Omega(\epsilon^2 m))$,*

$$|\widehat{F}_{TbV}(\eta) - F_{TbV}(\eta)| \leqslant \epsilon,$$

for all $\eta \in [0, 1/L]$.

In order to prove Lemma D.2.12, let's first show that for a fixed η with high probability $\widehat{F}_{TbV}(\eta)$ is close to $F_{TbV}(\eta)$. Similar as in Lemma D.2.8, we show each $\Delta_{TbV}(\eta, P_k)$ is $O(1)$-subexponential. We defer its proof to Section D.2.3.

Lemma D.2.17. *Suppose σ is a constant. For any fixed $\eta \in [0, 1/L]$ and any*

$1 > \epsilon > 0$, *with probability at least* $1 - \exp(-\Omega(\epsilon^2 m))$,

$$\left|\widehat{F}_{TbV}(\eta) - F_{TbV}(\eta)\right| \leqslant \epsilon.$$

Next, we show that there exists an ϵ-net for F_{TbV} with size $O(1/\epsilon)$. By ϵ-net, we mean there exists a finite set N_ϵ of step size such that $|F_{TbV}(\eta) - F_{TbV}(\eta')| \leqslant \epsilon$ for any $\eta \in [0, 1/L]$ and $\eta' \in \arg\min_{\eta \in N_\epsilon} |\eta - \eta'|$. We defer the proof of Lemma D.2.18 to Section D.2.3.

Lemma D.2.18. *Suppose σ is a constant. For any $1 > \epsilon > 0$, assume $d \geqslant c_4 \log(1/\epsilon)$ for constant c_4. There exists an ϵ-net N_ϵ for F_{TbV} with $|N_\epsilon| = O(1/\epsilon)$. That means, for any $\eta \in [0, 1/L]$,*

$$|F_{TbV}(\eta) - F_{TbV}(\eta')| \leqslant \epsilon,$$

for $\eta' \in \arg\min_{\eta \in N_\epsilon} |\eta - \eta'|$.

Next, we show that with high probability, there also exists an ϵ-net for $\widehat{F}_{TbV}$ with size $O(1/\epsilon)$.

Lemma D.2.19. *Suppose σ is a constant. For any $1 > \epsilon > 0$, assume $d \geqslant c_4 \log(1/\epsilon)$ for constant c_4. With probability at least $1 - \exp(-\Omega(\epsilon^2 m))$, there exists an ϵ-net N'_ϵ for $\widehat{F}_{TbV}$ with $|N_\epsilon| = O(1/\epsilon)$. That means, for any $\eta \in [0, 1/L]$,*

$$|\widehat{F}_{TbV}(\eta) - \widehat{F}_{TbV}(\eta')| \leqslant \epsilon,$$

for $\eta' \in \arg\min_{\eta \in N_\epsilon} |\eta - \eta'|$.

Combing Lemma D.2.17, Lemma D.2.18 and Lemma D.2.19, now we give the proof of Lemma D.2.12.

Proof of Lemma D.2.12. The proof is very similar as in Lemma D.2.4. By Lemma D.2.17, we know that with probability at least $1 - \exp(-\Omega(\epsilon^2 m))$, we have

$\left|\widehat{F}_{TbV}(\eta) - F_{TbV}(\eta)\right| \leqslant \epsilon$ for any fixed η. By Lemma D.2.18 and Lemma D.2.19, we know as long as $d = \Omega(\log(1/\epsilon))$, with probability at least $1 - \exp(-\Omega(\epsilon^2 m))$, there exists ϵ-net N_ϵ and N'_ϵ for F_{TbV} and $\widehat{F}_{TbV}$ respectively. Here, both of N_ϵ and N'_ϵ have size $O(1/\epsilon)$. According to the proofs of Lemma D.2.18 and Lemma D.2.19, it's not hard to verify that $N_\epsilon \cup N'_\epsilon$ is still an ϵ-net for $\widehat{F}_{TbV}$ and F_{TbV}. That means, for any $\eta \in [0, 1/L]$, we have

$$|F_{TbV}(\eta) - F_{TbV}(\eta')|, |\widehat{F}_{TbV}(\eta) - \widehat{F}_{TbV}(\eta')| \leqslant \epsilon,$$

for $\eta' \in \arg\min_{\eta \in N_\epsilon \cup N'_\epsilon} |\eta - \eta'|$.

Taking a union bound over $N_\epsilon \cup N'_\epsilon$, we have with probability at least $1 - O(1/\epsilon)\exp(-\Omega(\epsilon^2 m))$,

$$\left|\widehat{F}_{TbV}(\eta) - F_{TbV}(\eta)\right| \leqslant \epsilon$$

for any $\eta \in N_\epsilon \cup N'_\epsilon$.

Overall, we know with probability at least $1 - O(1/\epsilon)\exp(-\Omega(\epsilon^2 m))$, for all $\eta \in [0, 1/L]$,

$$\begin{aligned}
&|F_{TbV}(\eta) - \widehat{F}_{TbV}(\eta)| \\
\leqslant &|F_{TbV}(\eta) - F_{TbV}(\eta')| + |\widehat{F}_{TbV}(\eta) - \widehat{F}_{TbV}(\eta')| + |\widehat{F}_{TbV}(\eta') - F_{TbV}(\eta')| \\
\leqslant &3\epsilon,
\end{aligned}$$

where $\eta' \in \arg\min_{\eta \in N_\epsilon \cup N'_\epsilon} |\eta - \eta'|$. Changing ϵ to $\epsilon'/3$ finishes the proof. □

Lower Bounding $\widehat{F}_{TbV}$ for $\eta \in [1/L, \infty)$

In this section, we prove $\widehat{F}_{TbV}$ is large for any step size $\eta \geqslant 1/L$. Therefore, the optimal step size η^*_{valid} must be smaller than $\widehat{F}_{TbV}$.

Lemma D.2.13. *Suppose σ is a large constant. Assume $t \geqslant c_2, d \geqslant c_4 \log(t)$ for some constants c_2, c_4. With probability at least $1 - \exp(-\Omega(m))$,*

$$\widehat{F}_{TbV}(\eta) \geqslant C'\sigma^2 + \frac{1}{2}\sigma^2,$$

for all $\eta \geqslant 1/L$, where C' is a positive constant independent with σ.

When the step size is very large (larger than $3L$), we know the GD sequence gets truncated with high probability, which immediately implies the loss is high. The proof of Lemma D.2.20 is deferred into Section D.2.3.

Lemma D.2.20. *Assume $t \geqslant c_2, d \geqslant c_4$ for some constants c_2, c_4. With probability at least $1 - \exp(-\Omega(m))$,*

$$\widehat{F}_{TbV}(\eta) \geqslant \sigma^2,$$

for all $\eta \in [3L, \infty)$

The case for step size within $[1/L, 3L]$ requires more efforts. We give the proof of Lemma D.2.21 in this section later.

Lemma D.2.21. *Suppose σ is a large constant. Assume $t \geqslant c_2, d \geqslant c_4 \log(t)$ for some constants c_2, c_4. With probability at least $1 - \exp(-\Omega(m))$,*

$$\widehat{F}_{TbV}(\eta) \geqslant C_4\sigma^2 + \frac{1}{2}\sigma^2,$$

for all $\eta \in [1/L, 3L]$, where C_4 is a positive constant independent with σ.

With the above two lemmas, Lemma D.2.13 is just a combination of them.

Proof of Lemma D.2.13. The result follows by taking a union bound and choosing $C' = \min(C_4, 1/2)$. □

In the remaining of this section, we give the proof of Lemma D.2.21. When the step size is between $1/L$ and $3L$, if the GD sequence has a reasonable probability of

diverging, we can still show the loss is high similar as before. If not, we need to show the GD sequence overfits the noise in the training set, which incurs a high loss.

Recall that the noise term is roughly $\frac{1}{2}\left\|(I-(I-\eta H_{\text{train}})^t)(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2$. When $\eta \in [1/L, 3L]$, the eigenvalues of $I-\eta H_{\text{train}}$ in S_{train} subspace can be negative. If all the non-zero n eigenvalues of H_{train} have the same value, there exists a step size such that the eigenvalues of $I-\eta H_{\text{train}}$ in subspace S_{train} is -1. If t is even, the eigenvalues of $I-(I-\eta H_{\text{train}})^t$ in S_{train} subspace are zero, which means GD sequence does not catch any noise in S_{train}.

Notice that the above problematic case cannot happen when the eigenvalues of H_{train} are spread out. Basically, when there are two different eigenvalues, there won't exist any large η that can cancel both directions at the same time. In Lemma D.2.22, we show with constant probability, the eigenvalues of H_{train} are indeed spread out. The proof is deferred into Section D.2.3.

Lemma D.2.22. *Let the top n eigenvalues of H_{train} be $\lambda_1 \geqslant \cdots \geqslant \lambda_n$. Assume dimension $d \geqslant c_4$ for certain constant c_4. There exist positive constants μ, μ', μ'' such that with probability at least μ,*

$$\lambda_{\mu' n} - \lambda_{n-\mu' n+1} \geqslant \mu''.$$

Next, we utilize this variance in eigenvalues to prove that the GD sequence has to learn a constant fraction of the noise in training set.

Lemma D.2.23. *Suppose noise level σ is a large enough constant c_1. Assume unroll length $t \geqslant c_2$ and dimension $d \geqslant c_4$ for some constants c_2, c_4. Then, with probability at least C_1*

$$\|B_{t,\eta} w_{train} - w^*\|^2_{H_{train}} \geqslant C_2 \sigma^2,$$

for all $\eta \in [1/L, 3L]$*, where* C_1, C_2 *are positive constants.*

Proof of Lemma D.2.23. Let $\mathcal{E}_1$ be the event that $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\text{train}}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{\text{train}}\| \leqslant \sqrt{d}\sigma$. Let $\mathcal{E}_3$ be the event that $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\text{valid}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\text{valid}}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{\text{valid}}\| \leqslant \sqrt{d}\sigma$. According to Lemma D.2.3 and Lemma D.5.1, we know both $\mathcal{E}_1$ and $\mathcal{E}_3$ hold with probability at least $1 - \exp(-\Omega(d))$.

Let the top n eigenvalues of H_{train} be $\lambda_1 \geqslant \cdots \geqslant \lambda_n$. According to Lemma D.2.22, assuming d is larger than certain constant, we know there exist positive constants μ_1, μ_2, μ_3 such that with probability at least μ_1, $\lambda_{\mu_2 n} - \lambda_{n-\mu_2 n+1} \geqslant \mu_3$. Call this event $\mathcal{E}_2$.

Let S_1 and S_2 be the span of the bottom and top $\mu_2 n$ eigenvectors of H_{train} respectively. According to Lemma D.5.1, we know $\|\xi_{\text{train}}\| \geqslant \frac{\sqrt{d}}{4}\sigma$ with probability at least $1 - \exp(-\Omega(d))$. Let $P_1 \in \mathbb{R}^{n\times n}$ be a rank-$\mu_2 n$ projection matrix such that the column span of $(X_{\text{train}})^\dagger P_1$ is S_1. By Johnson-Lindenstrauss Lemma, we know with probability at least $1-\exp(-\Omega(d))$, $\left\|\text{Proj}_{P_1}\xi_{\text{train}}\right\| \geqslant \frac{\sqrt{\mu_2}}{2}\left\|\xi_{\text{train}}\right\|$. Taking a union bound, with probability at least $1-\exp(-\Omega(d))$, $\left\|\text{Proj}_{P_1}\xi_{\text{train}}\right\| \geqslant \frac{\sqrt{\mu_2 d}\sigma}{8}$. Similarly, we can define P_2 for the S_2 subspace and show with probability at least $1-\exp(-\Omega(d))$, $\left\|\text{Proj}_{P_2}\xi_{\text{train}}\right\| \geqslant \frac{\sqrt{\mu_2 d}\sigma}{8}$. Call the intersection of both events as $\mathcal{E}_4$, which happens with with probability at least $1 - \exp(-\Omega(d))$.

Taking a union bound, we know $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3 \cap \mathcal{E}_4$ holds with probability at least $\mu_1/2$ as long as d is larger than certain constant. Through the proof, we assume $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3 \cap \mathcal{E}_4$ holds.

Let's first lower bound $\|B_{t,\eta} w_{\text{train}} - w^*_{\text{train}}\|$ as follows,

$$\begin{aligned}\|B_{t,\eta} w_{\text{train}} - w^*_{\text{train}}\| &= \left\|B_{t,\eta}\left(w^*_{\text{train}} + (X_{\text{train}})^\dagger \xi_{\text{train}}\right) - w^*_{\text{train}}\right\| \\ &\geqslant \left(\left\|B_{t,\eta}\left(w^*_{\text{train}} + (X_{\text{train}})^\dagger \xi_{\text{train}}\right)\right\| - 1\right)\end{aligned}$$

Recall that we define S_1 and S_2 as the span of the bottom and top $\mu_2 n$ eigenvectors of H_{train} respectively. We rely on S_1 to lower bound $\|w_{t,\eta} - w^*\|$ when η is small and rely on S_2 when η is large.

Case 1: Let $\sigma^{S_1}_{\min}(B_{t,\eta})$ be the smallest singular value of $B_{t,\eta}$ within S_1 subspace. If $\eta\lambda_{n-\mu_2 n+1} \leqslant 2 - \mu_3/(2L)$, we have

$$\sigma^{S_1}_{\min}(B_{t,\eta}) \geqslant \min\left(1 - \left(1 - \frac{1}{L^2}\right)^t, 1 - \left(1 - \frac{\mu_3}{2L}\right)^t\right) \geqslant \frac{1}{2},$$

where the second inequality assumes $t \geqslant \max(L^2, 2L/\mu_3)\log 2$. Then, we have

$$\begin{aligned}\|w_{t,\eta} - w^*\| &\geqslant \left(\sigma^{S_1}_{\min}(B_{t,\eta})\left(\left\|\text{Proj}_{S_1}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\| - 1\right) - 1\right) \\ &\geqslant \left(\frac{1}{2}\left(\frac{\sqrt{\mu_2}\sigma}{8\sqrt{L}} - 1\right) - 1\right) \geqslant \frac{\sqrt{\mu_2}\sigma}{32\sqrt{L}},\end{aligned}$$

where the second inequality uses $\left\|\text{Proj}_{P_1}\xi_{\text{train}}\right\| \geqslant \frac{\sqrt{\mu_2 d}\sigma}{8}$ and the last inequality assumes $\sigma \geqslant \frac{48\sqrt{L}}{\sqrt{\mu_2}}$.

Case 2: If $\eta\lambda_{n-\mu_2 n+1} > 2 - \mu_3/(2L)$, we have $\eta\lambda_{\mu_2 n} \geqslant 2 + \mu_3/(2L)$ since $\lambda_{\mu_2 n} - \lambda_{n-\mu_2 n+1} \geqslant \mu_3$ and $\eta \geqslant 1/L$. Let $\sigma^{S_2}_{\min}(B_{t,\eta})$ be the smallest singular value of $B_{t,\eta}$ within S_2 subspace. We have

$$\sigma^{S_2}_{\min}(B_{t,\eta}) \geqslant \left(\left(1 + \frac{\mu_3}{2L}\right)^t - 1\right) \geqslant \frac{1}{2},$$

where the last inequality assumes $t \geqslant 4L/\mu_3$. Then, similar as in Case 1, we can also prove $\|w_{t,\eta} - w^*\| \geqslant \frac{\sqrt{\mu_2}\sigma}{32\sqrt{L}}$.

Therefore, we have

$$\|B_{t,\eta}w_{\text{train}} - w^*\|_{H_{\text{train}}}^2 = \|B_{t,\eta}w_{\text{train}} - w_{\text{train}}^*\|_{H_{\text{train}}}^2 \geqslant \frac{1}{L}\|B_{t,\eta}w_{\text{train}} - w_{\text{train}}^*\|^2 \geqslant \frac{\mu_2\sigma^2}{1024L^2},$$

for all $\eta \in [1/L, 3L]$. We denote $C_1 := \mu_1/2$ and $C_2 = \frac{\mu_2}{1024L^2}$. □

Before we present the proof of Lemma D.2.21, we still need a technical lemma that shows the noise in S_{valid} concentrates at its mean. The proof of Lemma D.2.24 is deferred into Section D.2.3.

Lemma D.2.24. *Suppose σ is constant. For any $1 > \epsilon > 0$, with probability at least $1 - O(t/\epsilon)\exp(-\Omega(\epsilon^2 d))$, $\lambda_n(H_{valid}) \geqslant 1/L$ and*

$$\|w_{t,\eta} - w_{valid}\|_{H_{valid}}^2 \geqslant \|w_{t,\eta} - w^*\|_{H_{valid}}^2 + (1-\epsilon)\sigma^2,$$

for all $\eta \in [1/L, 3L]$.

Combing the above lemmas, we give the proof of Lemma D.2.21.

Proof of Lemma D.2.21. According to Lemma D.2.24, we know given $1 > \epsilon > 0$, with probability at least
$1-O(t/\epsilon)\exp(-\Omega(\epsilon^2 d))$, $\lambda_n(H_{\text{valid}}) \geqslant 1/L$ and $\|w_{t,\eta} - w_{\text{valid}}\|_{H_{\text{valid}}}^2 \geqslant \|w_{t,\eta} - w^*\|_{H_{\text{valid}}}^2 +$ $(1-\epsilon)\sigma^2$ for all $\eta \in [1/L, 3L]$. Call this event $\mathcal{E}_1$. Suppose $\Pr[\mathcal{E}_1] \geqslant 1-\delta/2$, where δ will be specifies later. For each training set $S_{\text{train}}^{(k)}$, we also define $\mathcal{E}_1^{(k)}$. By concentration, we know with probability at least $1 - \exp(-\Omega(\delta^2 m))$, $1/m\sum_{k=1}^m \mathbb{1}\left\{\mathcal{E}_1^{(k)}\right\} \geqslant 1-\delta$.

According to Lemma D.2.23, we know there exist constants C_1, C_2 such that with probability at least C_1,
$\|B_{t,\eta}w_{\text{train}} - w^*\|_{H_{\text{train}}}^2 \geqslant C_2\sigma^2$ for all $\eta \in [1/L, 3L]$. Call this event $\mathcal{E}_2$. For each training set $S_{\text{train}}^{(k)}$, we also define $\mathcal{E}_2^{(k)}$. By concentration, we know with probability

at least $1-\exp(-\Omega(m))$, $1/m\sum_{k=1}^{m}\mathbb{1}\left\{\mathcal{E}_2^{(k)}\right\}\geqslant C_1/2$.

For any step size $\eta\in[1/L,3L]$, we can lower bound $\widehat{F}_{TbV}(\eta)$ as follows,

$$\begin{aligned}
\widehat{F}_{TbV}(\eta)=&\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)}-w_{\text{valid}}^{(k)}\right\|_{H_{\text{valid}}^{(k)}}^2\\
\geqslant&\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)}-w_{\text{valid}}^{(k)}\right\|_{H_{\text{valid}}^{(k)}}^2\mathbb{1}\left\{\mathcal{E}_1^{(k)}\right\}\\
\geqslant&\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)}-w_k^*\right\|_{H_{\text{valid}}}^2\mathbb{1}\left\{\mathcal{E}_1^{(k)}\right\}+\frac{1}{2}(1-\epsilon)(1-\delta)\sigma^2\\
\geqslant&\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)}-w_k^*\right\|_{H_{\text{valid}}}^2\mathbb{1}\left\{\mathcal{E}_1^{(k)}\cap\mathcal{E}_2^{(k)}\right\}+\frac{1}{2}(1-\epsilon)(1-\delta)\sigma^2.
\end{aligned}$$

As long as $\delta\leqslant C_1/4$, we know $\frac{1}{m}\sum_{k=1}^{m}\mathbb{1}\left\{\mathcal{E}_1^{(k)}\cap\mathcal{E}_2^{(k)}\right\}\geqslant C_1/4$. Let $\bar{\mathcal{E}}_3(\eta)$ be the event that $w_{t,\eta}^{(k)}$ gets truncated with step size η. We have

$$\begin{aligned}
&\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)}-w_k^*\right\|_{H_{\text{valid}}}^2\mathbb{1}\left\{\mathcal{E}_1^{(k)}\cap\mathcal{E}_2^{(k)}\right\}\\
=&\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)}-w_k^*\right\|_{H_{\text{valid}}}^2\mathbb{1}\left\{\mathcal{E}_1^{(k)}\cap\mathcal{E}_2^{(k)}\cap\mathcal{E}_3^{(k)}\right\}\\
&+\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)}-w_k^*\right\|_{H_{\text{valid}}}^2\mathbb{1}\left\{\mathcal{E}_1^{(k)}\cap\mathcal{E}_2^{(k)}\cap\bar{\mathcal{E}}_3^{(k)}\right\}.
\end{aligned}$$

If $\frac{1}{m}\sum_{k=1}^{m}\mathbb{1}\left\{\mathcal{E}_1^{(k)}\cap\mathcal{E}_2^{(k)}\cap\bar{\mathcal{E}}_3^{(k)}\right\}\geqslant C_1/8$, we have

$$\begin{aligned}
&\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)}-w_k^*\right\|_{H_{\text{valid}}}^2\mathbb{1}\left\{\mathcal{E}_1^{(k)}\cap\mathcal{E}_2^{(k)}\right\}\\
\geqslant&\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)}-w_k^*\right\|_{H_{\text{valid}}}^2\mathbb{1}\left\{\mathcal{E}_1^{(k)}\cap\mathcal{E}_2^{(k)}\cap\bar{\mathcal{E}}_3^{(k)}\right\}\\
\geqslant&\frac{C_1}{8}\times\frac{9\sigma^2}{2}=\frac{9C_1\sigma^2}{16}.
\end{aligned}$$

Here, we lower bound $\left\|w_{t,\eta}^{(k)}-w_k^*\right\|_{H_{\text{valid}}}^2$ by $9\sigma^2$ when the sequence gets truncated.

If $\frac{1}{m}\sum_{k=1}^{m}\mathbb{1}\left\{\mathcal{E}_1^{(k)}\cap\mathcal{E}_2^{(k)}\cap\bar{\mathcal{E}}_3^{(k)}\right\}<C_1/8$, we know $\frac{1}{m}\sum_{k=1}^{m}\mathbb{1}\left\{\mathcal{E}_1^{(k)}\cap\mathcal{E}_2^{(k)}\cap\mathcal{E}_3^{(k)}\right\}\geqslant C_1/8$. Then, we have

$$\begin{aligned}
&\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)}-w_k^*\right\|_{H_{\text{valid}}}^2\mathbb{1}\left\{\mathcal{E}_1^{(k)}\cap\mathcal{E}_2^{(k)}\right\}\\
\geqslant&\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|B_{t,\eta}^{(k)}w_{\text{train}}-w_k^*\right\|_{H_{\text{valid}}}^2\mathbb{1}\left\{\mathcal{E}_1^{(k)}\cap\mathcal{E}_2^{(k)}\cap\mathcal{E}_3^{(k)}\right\}\\
\geqslant&\frac{C_1}{8}\times\frac{C_2\sigma^2}{2}=\frac{C_1C_2\sigma^2}{16}
\end{aligned}$$

Letting $C_3=\min(\frac{9C_1}{16},\frac{C_1C_2}{16})$, we then have

$$\widehat{F}_{TbV}(\eta)\geqslant C_3\sigma^2+\frac{1}{2}(1-\epsilon)(1-\delta)\sigma^2\geqslant\frac{C_3\sigma^2}{2}+\frac{1}{2}\sigma^2,$$

where the last inequality chooses $\delta=\epsilon=C_3/2$. In order for $\Pr[\mathcal{E}_1]\geqslant 1-\delta/2$, we only need $d\geqslant c_4\log(t)$ for some constant c_4. Replacing $C_3/2$ by C_4 finishes the proof. □

Proofs of Technical Lemmas

Proof of Lemma D.2.15. We first show that for a fixed $\eta\in[0,1/L]$, the crossing term $\left|\langle B_{t,\eta}w_{\text{train}}^*-w^*\rangle\, B_{t,\eta}(X_{\text{train}})^\dagger\xi_{\text{train}}\right|$ is small with high probability. We can write

down the crossing term as follows:

$$\langle B_{t,\eta} w_{\text{train}}^* - w^* \rangle B_{t,\eta} (X_{\text{train}})^\dagger \xi_{\text{train}} = \big\langle [(X_{\text{train}})^\dagger]^\top B_{t,\eta} (B_{t,\eta} w_{\text{train}}^* - w^*) \big\rangle \xi_{\text{train}}.$$

Noticing that ξ_{train} is independent with $[(X_{\text{train}})^\dagger]^\top B_{t,\eta}(B_{t,\eta} w_{\text{train}}^* - w^*)$, we will use Hoeffding's inequality to bound $\big|\langle B_{t,\eta} w_{\text{train}}^* - w^* \rangle B_{t,\eta} (X_{\text{train}})^\dagger \xi_{\text{train}}\big|$. According to Lemma D.2.3, we know with probability at least $1 - \exp(-\Omega(d))$, $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\text{train}}) \leqslant L$ for all $i \in [n]$ with $L = 100$. Since $\eta \leqslant 1/L$, we know $\|B_{t,\eta}\| = \|I - (I - \eta H_{\text{train}})^t\| \leqslant 1$. Therefore, we have

$$\big\| [(X_{\text{train}})^\dagger]^\top B_{t,\eta} (B_{t,\eta} w_{\text{train}}^* - w^*) \big\| \leqslant \frac{2\sqrt{L}}{\sqrt{d}},$$

for any $\eta \in [0, 1/L]$. Then, for any $\epsilon > 0$, by Hoeffding's inequality, with probability at least $1 - \exp(-\Omega(\epsilon^2 d))$,

$$\big|\langle B_{t,\eta} w_{\text{train}}^* - w^* \rangle B_{t,\eta} (X_{\text{train}})^\dagger \xi_{\text{train}}\big| \leqslant \epsilon.$$

Next, we construct an ϵ-net on η and show the crossing term is small for all $\eta \in [0, 1/L]$. Let

$$g(\eta) := \langle B_{t,\eta} w_{\text{train}}^* - w^* \rangle B_{t,\eta} (X_{\text{train}})^\dagger \xi_{\text{train}}.$$

We compute the derivative of $g(\eta)$ as follows:

$$\begin{aligned} g'(\eta) = & \big\langle t H_{\text{train}} (I - \eta H_{\text{train}})^{t-1} w_{\text{train}}^* \big\rangle B_{t,\eta} (X_{\text{train}})^\dagger \xi_{\text{train}} \\ & + \langle B_{t,\eta} w_{\text{train}}^* - w^* \rangle t H_{\text{train}} (I - \eta H_{\text{train}})^{t-1} (X_{\text{train}})^\dagger \xi_{\text{train}} \end{aligned}$$

By Lemma D.5.1, we know with probability at least $1 - \exp(-\Omega(d))$, $\|\xi_{\text{train}}\| \leqslant \sqrt{d}\sigma$. Therefore,

$$|g'(\eta)| \leqslant L^{1.5} t \left(1 - \frac{\eta}{L}\right)^{t-1} \sigma + 2L^{1.5} t \left(1 - \frac{\eta}{L}\right)^{t-1} \sigma = 3L^{1.5} t \left(1 - \frac{\eta}{L}\right)^{t-1} \sigma.$$

We can control $|g'(\eta)|$ in different regimes:

- For $\eta \in [0, \frac{L}{t-1}]$, we have $|g'(\eta)| \leqslant 3L^{1.5}t\sigma$.
- Given any $1 \leqslant i \leqslant \log t - 1$, for any $\eta \in (\frac{iL}{t-1}, \frac{(i+1)L}{t-1}]$, we have $|g'(\eta)| \leqslant \frac{3L^{1.5}t\sigma}{e^i}$.
- For any $\eta \in (\frac{L\log t}{t-1}, 1/L]$, we have $|g'(\eta)| \leqslant 3L^{1.5}\sigma$.

Fix any $\epsilon > 0$, we know there exists an ϵ-net N_ϵ with size

$$\begin{aligned}|N_\epsilon| =& \frac{1}{\epsilon}\left(\frac{L}{t-1}\sum_{i=0}^{\log t-1}\frac{3L^{1.5}t\sigma}{e^i} + \left(\frac{1}{L} - \frac{L\log t}{t-1}\right)3L^{1.5}\sigma\right)\\ \leqslant& \frac{1}{\epsilon}\left(\frac{3eL^{2.5}t\sigma}{t-1} + 3\sqrt{L}\sigma\right) = O(\frac{1}{\epsilon})\end{aligned}$$

such that for any $\eta \in [0, 1/L]$, there exists $\eta' \in N_\epsilon$ with $|g(\eta) - g(\eta')| \leqslant \epsilon$. Note that $L = 100$ and σ is a constant. Taking a union bound over N_ϵ and all the other bad events, we have with probability at least $1 - \exp(-\Omega(d)) - O(1/\epsilon)\exp(-\Omega(\epsilon^2 d))$, for all $\eta \in [0, 1/L]$,

$$\left|\left\langle B_{t,\eta}w^*_{\text{train}} - w^*\right\rangle B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right| \leqslant \epsilon + \epsilon = 2\epsilon.$$

As long as $1 > \epsilon > 0$, this happens with probability at least $1 - O(1/\epsilon)\exp(-\Omega(\epsilon^2 d))$. Replacing ϵ by $\epsilon'/2$ finishes the proof. □

Proof of Lemma D.2.16. According to Lemma D.2.3, we know with probability at least $1 - \exp(-\Omega(d))$, $1/L \leqslant \lambda_i(H_{\text{train}}) \leqslant L$ for all $i \in [n]$ with $L = 100$. We can lower bound $\langle B_{t,\eta}w^*_{\text{train}}\rangle w^*$ as follows,

$$\begin{aligned}\left\langle B_{t,\eta}w^*_{\text{train}}\right\rangle w^* =& \left\langle \left(I - (I - \eta H_{\text{train}})^t\right) w^*_{\text{train}}\right\rangle w^*_{\text{train}}\\ \geqslant& \lambda_{\min}\left(I - (I - \eta H_{\text{train}})^t\right)\left\|w^*_{\text{train}}\right\|^2\\ \geqslant& \left(1 - \exp\left(-\frac{\eta t}{L}\right)\right)\left\|w^*_{\text{train}}\right\|^2.\end{aligned}$$

By Johnson-Lindenstrauss lemma (Lemma D.5.5), we know with probability at least $1 - 2\exp(-c\epsilon^2 d/4)$,

$$\|w^*_{\text{train}}\| \geqslant \frac{1}{2}(1-\epsilon)\|w^*\| = \frac{1}{2}(1-\epsilon).$$

Then, we know with probability at least $1 - 2\exp(-c\epsilon^2 d/4) - \exp(-\Omega(d))$,

$$\begin{aligned}\langle B_{t,\eta} w^*_{\text{train}} \rangle w^* &\geqslant \left(1 - \exp\left(-\frac{\eta t}{L}\right)\right)\|w^*_{\text{train}}\|^2 \\ &\geqslant \left(1 - \exp\left(-\frac{\eta t}{L}\right)\right)\frac{1}{4}(1-\epsilon)^2 \\ &\geqslant \frac{1-2\epsilon}{4}\left(1 - \exp\left(-\frac{\eta t}{L}\right)\right)\end{aligned}$$

Since $e^x \leqslant 1 - x + x^2/2$ for any $x \leqslant 0$, we know $\exp(-\eta t/L) \leqslant 1 - \eta t/L + \eta^2 t^2/(2L^2)$. For any $\eta \leqslant L/t$, we have $\exp(-\eta t/L) \leqslant 1 - \eta t/(2L)$. Then with probability at least $1 - 2\exp(-c\epsilon^2 d/4) - \exp(-\Omega(d))$,

$$\begin{aligned}\langle B_{t,\eta} w^*_{\text{train}} \rangle w^* &\geqslant \frac{1-2\epsilon}{4}\frac{\eta t}{2L} \\ &\geqslant \frac{\eta t}{16L},\end{aligned}$$

where the second inequality holds by choosing $\epsilon = 1/4$. □

Proof of Lemma D.2.17. Recall that

$$\widehat{F}_{TbV}(\eta) := \frac{1}{m}\sum_{k=1}^{m} \Delta_{TbV}(\eta, P_k)$$

For each individual loss function $\Delta_{TbV}(\eta, P_k)$, we have

$$\begin{aligned}\Delta_{TbV}(\eta, P_k) =& \frac{1}{2}\left\|w_{t,\eta}^{(k)} - w^* - (X_{\text{valid}}^{(k)})^\dagger \xi_{\text{valid}}^{(k)}\right\|_{H_{\text{valid}}^{(k)}}^2 \\ =& \frac{1}{2}\left\|w_{t,\eta}^{(k)} - w^*\right\|_{H_{\text{valid}}^{(k)}}^2 + \frac{1}{2n}\left\|\xi_{\text{valid}}^{(k)}\right\|^2 + \left\langle w_{t,\eta}^{(k)} - w^* \right\rangle \frac{1}{n}(X_{\text{valid}}^{(k)})^\top \xi_{\text{valid}}^{(k)} \\ \leqslant& \frac{25L\sigma^2}{2}\left\|H_{\text{valid}}^{(k)}\right\| + \frac{1}{2n}\left\|\xi_{\text{valid}}^{(k)}\right\|^2 + 5\sqrt{L}\sigma\left(\frac{1}{\sqrt{n}}\left\|X_{\text{valid}}^{(k)}\right\|\right)\left(\frac{1}{\sqrt{n}}\left\|\xi_{\text{valid}}^{(k)}\right\|\right)\end{aligned}$$

We can write $\left\|H_{\text{valid}}^{(k)}\right\|$ as $\sigma_{\max}^2(\frac{1}{\sqrt{n}}X_{\text{valid}}^{(k)})$. According to Lemma D.5.3, we know $\sigma_{\max}(X_{\text{valid}}^{(k)}) - \mathbb{E}\sigma_{\max}(X_{\text{valid}}^{(k)})$ is $O(1)$-subgaussian, which implies that $\sigma_{\max}(\frac{1}{\sqrt{n}}X_{\text{valid}}^{(k)}) - \mathbb{E}\sigma_{\max}(\frac{1}{\sqrt{n}}X_{\text{valid}}^{(k)})$ is $O(1/\sqrt{d})$-subgaussian. Since $\mathbb{E}\sigma_{\max}(\frac{1}{\sqrt{n}}X_{\text{valid}}^{(k)})$ is a constant, we know $\sigma_{\max}(\frac{1}{\sqrt{n}}X_{\text{valid}}^{(k)})$ is $O(1)$-subgaussian and $\sigma_{\max}^2(\frac{1}{\sqrt{n}}X_{\text{valid}}^{(k)})$ is $O(1)$-subexponential. Similarly, we know both the terms $\frac{1}{2n}\left\|\xi_{\text{valid}}^{(k)}\right\|^2$ and $\left(\frac{1}{\sqrt{n}}\left\|X_{\text{valid}}^{(k)}\right\|\right)\left(\frac{1}{\sqrt{n}}\left\|\xi_{\text{valid}}^{(k)}\right\|\right)$ are $O(1)$-subexponential. This further implies that $\Delta_{TbV}(\eta, P_k)$ is $O(1)$-subexponential. Therefore, $\widehat{F}_{TbV}$ is the average of m i.i.d. $O(1)$-subexponential random variables. By standard concentration inequality, we know for any $1 > \epsilon > 0$, with probability at least $1 - \exp(-\Omega(\epsilon^2 m))$,

$$\left|\widehat{F}_{TbV}(\eta) - F_{TbV}(\eta)\right| \leqslant \epsilon.$$

□

Proof of Lemma D.2.18. Recall that

$$F_{TbV}(\eta) = \mathbb{E}\frac{1}{2}\|w_{t,\eta} - w^*\|^2 + \sigma^2/2.$$

We only need to construct an ϵ-net for $\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w^*\|^2$. Let $\mathcal{E}$ be the event that $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\text{train}}) \leqslant L$ for all $i \in [n]$ and $\|\xi_{\text{train}}\| \leqslant$

$\sqrt{d}\sigma$. We have

$$\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w^*\|^2 = \mathbb{E}\left[\frac{1}{2}\|w_{t,\eta} - w^*\|^2|\mathcal{E}\right]\Pr[\mathcal{E}] + \mathbb{E}\left[\frac{1}{2}\|w_{t,\eta} - w^*\|^2|\bar{\mathcal{E}}\right]\Pr[\bar{\mathcal{E}}]$$

We first construct ϵ-net for $\mathbb{E}\left[\frac{1}{2}\|w_{t,\eta} - w^*\|^2|\mathcal{E}\right]\Pr[\mathcal{E}]$. Let $Q(\eta) := \frac{1}{2}\|w_{t,\eta} - w^*\|^2$. Fix a training set S_{train} under which event $\mathcal{E}$ holds. We show that $Q(\eta)$ has desirable lipschitz property.

The derivative of $Q(\eta)$ can be computed as follows,

$$Q'(\eta) = \left\langle tH_{\text{train}}(I - \eta H_{\text{train}})^{t-1}w_{\text{train}}\right\rangle w_{t,\eta} - w^*.$$

Conditioning on $\mathcal{E}$, we have

$$|Q'(\eta)| = O(1)t(1 - \frac{\eta}{L})^{t-1}.$$

Therefore, we have

$$\left|\frac{\partial}{\partial\eta}\mathbb{E}\left[\frac{1}{2}\|w_{t,\eta} - w^*\|^2|\mathcal{E}\right]\Pr[\mathcal{E}]\right| = O(1)t(1 - \frac{\eta}{L})^{t-1}.$$

Similar as in Lemma D.2.15, for any $\epsilon > 0$, we know there exists an ϵ-net N_ϵ with size $O(1/\epsilon)$ such that for any $\eta \in [0, 1/L]$,

$$\left|\mathbb{E}\left[\frac{1}{2}\|w_{t,\eta} - w^*\|^2|\mathcal{E}\right]\Pr[\mathcal{E}] - \mathbb{E}\left[\frac{1}{2}\|w_{t,\eta'} - w^*\|^2|\mathcal{E}\right]\Pr[\mathcal{E}]\right| \leqslant \epsilon$$

for $\eta' \in \arg\min_{\eta\in N_\epsilon}|\eta - \eta'|$.

Suppose the probability of $\bar{\mathcal{E}}$ is δ. We have

$$\mathbb{E}\left[\frac{1}{2}\|w_{t,\eta} - w^*\|^2|\bar{\mathcal{E}}\right]\Pr[\bar{\mathcal{E}}] \leqslant \frac{25L\sigma^2}{2}\delta \leqslant \epsilon,$$

where the last inequality assumes $\delta \leqslant \frac{2\epsilon}{25L\sigma^2}$. From Lemma D.2.3 and Lemma D.5.1,

we know $\delta := \Pr\left[\bar{\mathcal{E}}\right] \leqslant \exp(-\Omega(d))$. Therefore, given any $\epsilon > 0$, there exists constant c_4 such that $\delta \leqslant \frac{2\epsilon}{25L\sigma^2}$ as long as $d \geqslant c_4 \log(1/\epsilon)$.

Overall, for any $\epsilon > 0$, as long as $d = \Omega(\log(1/\epsilon))$, there exists N_ϵ with size $O(1/\epsilon)$ such that for any $\eta \in [0, 1/L]$, $|F_{TbV}(\eta) - F_{TbV}(\eta')| \leqslant 3\epsilon$ for $\eta' \in \arg\min_{\eta \in N_\epsilon} |\eta - \eta'|$. Changing ϵ to $\epsilon'/3$ finishes the proof. □

Proof of Lemma D.2.19. For each $k \in [m]$, let $\mathcal{E}_k$ be the event that $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X^{(k)}_{\text{train}}) \leqslant \sqrt{Ld}$ for any $i \in [n]$ and $\left\|\xi^{(k)}_{\text{train}}\right\| \leqslant \sqrt{d}\sigma$. Then, we can write the empirical meta objective as follows,

$$\widehat{F}_{TbV}(\eta) := \frac{1}{m}\sum_{k=1}^{m} \Delta_{TbT}(\eta, P_k)\mathbb{1}_{\mathcal{E}_k} + \frac{1}{m}\sum_{k=1}^{m} \Delta_{TbT}(\eta, P_k)\mathbb{1}_{\bar{\mathcal{E}}_k}.$$

Similar as Lemma D.2.18, we will show that the first term has desirable Lipschitz property and the second term is small. Now, let's focus on the first term $\frac{1}{m}\sum_{k=1}^{m} \Delta_{TbT}(\eta, P_k)\mathbb{1}_{\mathcal{E}_k}$. Recall that

$$\begin{aligned}\Delta_{TbT}(\eta, P_k) =& \frac{1}{2}\left\|w^{(k)}_{t,\eta} - w^{(k)}_{\text{valid}}\right\|^2_{H^{(k)}_{\text{valid}}} \\ =& \frac{1}{2}\left\|B^{(k)}_{t,\eta} w^{(k)}_{\text{train}} - w^* - (X^{(k)}_{\text{valid}})^\dagger \xi^{(k)}_{\text{valid}}\right\|^2_{H^{(k)}_{\text{valid}}}.\end{aligned}$$

Computing the derivative of $\Delta_{TbT}(\eta, P_k)$ in terms of η, we have

$$\frac{\partial}{\partial\eta}\Delta_{TbT}(\eta, P_k) = \left\langle tH^{(k)}_{\text{train}}(I - \eta H^{(k)}_{\text{train}})^{t-1} w^{(k)}_{\text{train}} \right\rangle H^{(k)}_{\text{valid}}\left(w^{(k)}_{t,\eta} - w^* - (X^{(k)}_{\text{valid}})^\dagger \xi^{(k)}_{\text{valid}}\right)$$

Conditioning on $\mathcal{E}_k$, we can bound the derivative,

$$\left|\frac{\partial}{\partial\eta}\Delta_{TbT}(\eta, P_k)\right| = O(1)t\left(1 - \frac{\eta}{L}\right)^{t-1}\left(\left\|H^{(k)}_{\text{valid}}\right\| + \left(\frac{1}{\sqrt{d}}\left\|X^{(k)}_{\text{valid}}\right\|\right)\left(\frac{1}{\sqrt{d}}\left\|\xi^{(k)}_{\text{valid}}\right\|\right)\right).$$

Therefore, we have

$$\left|\frac{1}{m}\sum_{k=1}^{m}\frac{\partial}{\partial\eta}\Delta_{TbT}(\eta,P_k)\mathbb{1}_{\mathcal{E}_k}\right|$$

$$=O(1)t\left(1-\frac{\eta}{L}\right)^{t-1}\frac{1}{m}\sum_{k=1}^{m}\left(\left\|H_{\text{valid}}^{(k)}\right\|+\left(\frac{1}{\sqrt{d}}\left\|X_{\text{valid}}^{(k)}\right\|\right)\left(\frac{1}{\sqrt{d}}\left\|\xi_{\text{valid}}^{(k)}\right\|\right)\right).$$

Similar as in Lemma D.2.17, we know both $\left\|H_{\text{valid}}^{(k)}\right\|$ and $\left(\frac{1}{\sqrt{d}}\left\|X_{\text{valid}}^{(k)}\right\|\right)\left(\frac{1}{\sqrt{d}}\left\|\xi_{\text{valid}}^{(k)}\right\|\right)$ are $O(1)$-subexponential. Therefore, we know with probability at least $1-\exp(-\Omega(m))$, $\frac{1}{m}\sum_{k=1}^{m}\left(\left\|H_{\text{valid}}^{(k)}\right\|+\left(\frac{1}{\sqrt{d}}\left\|X_{\text{valid}}^{(k)}\right\|\right)\left(\frac{1}{\sqrt{d}}\left\|\xi_{\text{valid}}^{(k)}\right\|\right)\right)=O(1)$. This further shows that with probability at least $1-\exp(-\Omega(m))$,

$$\left|\frac{1}{m}\sum_{k=1}^{m}\frac{\partial}{\partial\eta}\Delta_{TbT}(\eta,P_k)\mathbb{1}_{\mathcal{E}_k}\right|=O(1)t\left(1-\frac{\eta}{L}\right)^{t-1}.$$

Similar as in Lemma D.2.15, we can show that for any $\epsilon>0$, there exists an ϵ-net with size $O(1/\epsilon)$ for $\frac{1}{m}\sum_{k=1}^{m}\Delta_{TbT}(\eta,P_k)\mathbb{1}_{\mathcal{E}_k}$.

Next, we show that the second term $\frac{1}{m}\sum_{k=1}^{m}\Delta_{TbT}(\eta,P_k)\mathbb{1}_{\bar{\mathcal{E}}_k}$ is small with high probability. According to the proof in Lemma D.2.17, we know

$$\Delta_{TbT}(\eta,P_k)=O(1)\left(\left\|H_{\text{valid}}^{(k)}\right\|+\frac{1}{d}\left\|\xi_{\text{valid}}^{(k)}\right\|^2+\left(\frac{1}{\sqrt{d}}\left\|X_{\text{valid}}^{(k)}\right\|\right)\left(\frac{1}{\sqrt{d}}\left\|\xi_{\text{valid}}^{(k)}\right\|\right)\right)$$

Therefore, there exists constant C such that

$$\frac{1}{m}\sum_{k=1}^{m}\Delta_{TbT}(\eta,P_k)\mathbb{1}_{\bar{\mathcal{E}}_k}$$

$$\leqslant C\frac{1}{m}\sum_{k=1}^{m}\left(\left\|H_{\text{valid}}^{(k)}\right\|+\frac{1}{d}\left\|\xi_{\text{valid}}^{(k)}\right\|^2+\left(\frac{1}{\sqrt{d}}\left\|X_{\text{valid}}^{(k)}\right\|\right)\left(\frac{1}{\sqrt{d}}\left\|\xi_{\text{valid}}^{(k)}\right\|\right)\right)\mathbb{1}_{\bar{\mathcal{E}}_k}.$$

It's not hard to verify that $\left(\left\|H_{\text{valid}}^{(k)}\right\|+\frac{1}{d}\left\|\xi_{\text{valid}}^{(k)}\right\|^2+\left(\frac{1}{\sqrt{d}}\left\|X_{\text{valid}}^{(k)}\right\|\right)\left(\frac{1}{\sqrt{d}}\left\|\xi_{\text{valid}}^{(k)}\right\|\right)\right)\mathbb{1}_{\bar{\mathcal{E}}_k}$

is $O(1)$-subexponential.

Suppose the expectation of $\left(\left\|H^{(k)}_{\text{valid}}\right\| + \frac{1}{d}\left\|\xi^{(k)}_{\text{valid}}\right\|^2 + \left(\frac{1}{\sqrt{d}}\left\|X^{(k)}_{\text{valid}}\right\|\right)\left(\frac{1}{\sqrt{d}}\left\|\xi^{(k)}_{\text{valid}}\right\|\right)\right)$ is μ, which is a constant. Suppose the probability of $\bar{\mathcal{E}_k}$ be δ. We know the expectation of $\left(\left\|H^{(k)}_{\text{valid}}\right\| + \frac{1}{d}\left\|\xi^{(k)}_{\text{valid}}\right\|^2 + \left(\frac{1}{\sqrt{d}}\left\|X^{(k)}_{\text{valid}}\right\|\right)\left(\frac{1}{\sqrt{d}}\left\|\xi^{(k)}_{\text{valid}}\right\|\right)\right)\mathbb{1}_{\bar{\mathcal{E}_k}}$ is $\mu\delta$ due to independence. By standard concentration inequality, for any $1 > \epsilon > 0$, with probability at least $1 - \exp(-\Omega(\epsilon^2 m))$,

$$
\begin{aligned}
&C\frac{1}{m}\sum_{k=1}^{m}\left(\left\|H^{(k)}_{\text{valid}}\right\| + \frac{1}{d}\left\|\xi^{(k)}_{\text{valid}}\right\|^2 + \left(\frac{1}{\sqrt{d}}\left\|X^{(k)}_{\text{valid}}\right\|\right)\left(\frac{1}{\sqrt{d}}\left\|\xi^{(k)}_{\text{valid}}\right\|\right)\right)\mathbb{1}_{\bar{\mathcal{E}_k}} \\
\leqslant & C\mu\delta + C\epsilon \leqslant (C+1)\epsilon,
\end{aligned}
$$

where the second inequality assumes $\delta \leqslant \epsilon/(C\mu)$. By Lemma D.2.3 and Lemma D.5.1, we know $\delta \leqslant \exp(-\Omega(d))$. Therefore, as long as $d \geqslant c_4 \log(1/\epsilon)$ for some constant c_4, we have $\delta \leqslant \epsilon/(C\mu)$.

Overall, we know that as long as $d \geqslant c_4 \log(1/\epsilon)$, with probability at least $1 - \exp(-\Omega(\epsilon^2 m))$, there exists N'_ϵ with $|N'_\epsilon| = O(1/\epsilon)$ such that for any $\eta \in [0, 1/L]$,

$$
|\widehat{F}_{TbV}(\eta) - \widehat{F}_{TbV}(\eta')| \leqslant (2C+3)\epsilon,
$$

for $\eta' \in \arg\min_{\eta \in N_\epsilon} |\eta - \eta'|$. Changing ϵ to $\epsilon'/(2C+3)$ finishes the proof. □

Proof of Lemma D.2.20. Let $\mathcal{E}_1$ be the event that $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\text{train}}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{\text{train}}\| \leqslant \sqrt{d}\sigma$. Let $\mathcal{E}_2$ be the event that $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\text{valid}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\text{valid}}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{\text{valid}}\| \leqslant \sqrt{d}\sigma$. According to Lemma D.2.3 and Lemma D.5.1, we know both $\mathcal{E}_1$ and $\mathcal{E}_2$ hold with probability at least $1 - \exp(-\Omega(d))$. Assuming $d \geqslant c_4$ for certain constant c_4, we know $\Pr[\mathcal{E}_1 \cap \mathcal{E}_2] \geqslant 2/3$. Also define $\mathcal{E}_1^{(k)}$ and

$\mathcal{E}_2^{(k)}$ on each training set $S_{\text{train}}^{(k)}$. By concentration, we know with probability at least $1 - \exp(-\Omega(m))$,

$$\frac{1}{m}\sum_{k=1}^{m} \mathbb{1}\left\{\mathcal{E}_1^{(k)} \cap \mathcal{E}_2^{(k)}\right\} \geqslant \frac{1}{2}.$$

It's easy to verify that conditioning on $\mathcal{E}_1$, the GD sequence always exceeds the norm threshold and gets truncated for $\eta \geqslant 3L$ as long as t is larger than certain constant. We can lower bound $\widehat{F}_{TbV}$ for any $\eta \geqslant 3L$ as follows,

$$\begin{aligned}
\widehat{F}_{TbV}(\eta) =& \frac{1}{m}\sum_{k=1}^{m} \frac{1}{2}\left\|w_{t,\eta}^{(k)} - w_{\text{valid}}^{(k)}\right\|_{H_{\text{valid}}^{(k)}}^2 \\
\geqslant& \frac{1}{m}\sum_{k=1}^{m} \frac{1}{2}\left\|w_{t,\eta}^{(k)} - w_{\text{valid}}^{(k)}\right\|_{H_{\text{valid}}^{(k)}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_2\right\} \geqslant 2\sigma^2\frac{1}{2} = \sigma^2,
\end{aligned}$$

where the last inequality lower bounds $\left\|w_{t,\eta}^{(k)} - w_{\text{valid}}^{(k)}\right\|_{H_{\text{valid}}^{(k)}}^2$ by $2\sigma^2$ when $w_{t,\eta}^{(k)}$ gets truncated. □

Proof of Lemma D.2.22. We first show that with constant probability in X_{train}, the variance of the eigenvalues of H_{train} is lower bounded by a constant. Let $\bar{\lambda}$ be $1/n\sum_{i=1}^n \lambda_i$. Specifically, we show $1/n\sum_{i=1}^n \lambda_i^2 - \bar{\lambda}^2$ is lower bounded by a constant.

Let's first compute the variance of the eigenvalues in expectation. Let the i-th row of X_{train} be $x_i^\top$. We have,

$$\begin{aligned}
\mathbb{E}\left[\bar{\lambda}^2\right] = \frac{1}{n^2}\mathbb{E}\left[\left(\operatorname{tr}\left(\frac{1}{n}X_{\text{train}}^\top X_{\text{train}}\right)\right)^2\right] =& \frac{1}{n^4}\mathbb{E}\left[\left(\sum_{i=1}^n \|x_i\|^2\right)^2\right] \\
=& \frac{1}{n^4}\sum_{i=1}^n \mathbb{E}\,\|x_i\|^4 + \frac{1}{n^4}\sum_{1\leqslant i\neq j\leqslant n} \mathbb{E}\|x_i\|^2\|x_j\|^2 \\
=& \frac{1}{n^4}\left(nd(d+2) + n(n-1)d^2\right) = \frac{d^2}{n^2} + \frac{2d}{n^3}.
\end{aligned}$$

Similarly, we compute $\mathbb{E}\left[1/n\sum_{i=1}^n \lambda_i^2\right]$ as follows,

$$\begin{aligned}\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \lambda_i^2\right] =& \frac{1}{n^3}\mathbb{E}\left[\mathrm{tr}\left(X_{\mathrm{train}}^\top X_{\mathrm{train}} X_{\mathrm{train}}^\top X_{\mathrm{train}}\right)\right] \\ =& \frac{1}{n^3}\sum_{i=1}^n \mathbb{E}\left\|x_i\right\|^4 + \frac{1}{n^3}\sum_{1\leqslant i\neq j\leqslant n} \mathbb{E}\left\langle x_i\right\rangle {x_j}^2 \\ =& \frac{1}{n^3}\left(nd(d+2)+n(n-1)d\right) = \frac{d^2}{n^2}+\frac{d}{n}+\frac{d}{n^2}\end{aligned}$$

Therefore, we have

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \lambda_i^2 - \bar{\lambda}^2\right] = \frac{d}{n}+\frac{d}{n^2}-\frac{2d}{n^3} \geqslant \frac{d}{n} \geqslant \frac{4}{3},$$

where the first inequality assumes $n \geqslant 2$ and the last inequality uses $n \leqslant \frac{3d}{4}$. Since $n \geqslant \frac{1}{4}d$, we know $n \geqslant 2$ as long as $d \geqslant 8$.

Let $\mathcal{E}$ be the event that $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\mathrm{train}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\mathrm{train}}) \leqslant L$ for $i \in [n]$ with $L = 100$. According to Lemma D.2.3, we know $\mathcal{E}$ happens with probability at least $1-\exp(-\Omega(d))$. Let $\mathbb{1}\left\{\mathcal{E}\right\}$ be the indicator function for event $\mathcal{E}$. Next we show that $\mathbb{E}[1/n\sum_{i=1}^n(\lambda_i-\bar{\lambda})^2\mathbb{1}\left\{\mathcal{E}\right\}]$ is also lower bounded.

It's clear that $\mathbb{E}\left[\bar{\lambda}^2\mathbb{1}\left\{\mathcal{E}\right\}\right]$ is upper bounded by $\mathbb{E}\left[\bar{\lambda}^2\right]$. In order to lower bound $\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \lambda_i^2\mathbb{1}\left\{\mathcal{E}\right\}\right]$, we first show that $\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \lambda_i^2\mathbb{1}\left\{\bar{\mathcal{E}}\right\}\right]$ is small. We can decompose $\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \lambda_i^2\mathbb{1}\left\{\bar{\mathcal{E}}\right\}\right]$ into two parts,

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \lambda_i^2\mathbb{1}\left\{\bar{\mathcal{E}}\right\}\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \lambda_i^2\mathbb{1}\left\{\bar{\mathcal{E}} \text{ and } \lambda_1 \leqslant L\right\}\right] + \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \lambda_i^2\mathbb{1}\left\{\lambda_1 > L\right\}\right].$$

The first term can be bounded by $L^2\Pr\left[\bar{\mathcal{E}}\right]$. Since $\Pr\left[\bar{\mathcal{E}}\right] \leqslant \exp(-\Omega(d))$, we know the first term is at most $1/6$ as long as d is larger than certain constant. The second term can be bounded by $\mathbb{E}\left[\lambda_1^2\mathbb{1}\left\{\lambda_1 > L\right\}\right]$. According to Lemma D.5.4, we know

$\Pr[\lambda_1 \geqslant L + t] \leqslant \exp(-\Omega(dt))$. Then, it's not hard to verify that $\mathbb{E}\left[\lambda_1^2 \mathbb{1}\left\{\lambda_1 > L\right\}\right] = O(1/d)$ that is bounded by $1/6$ as long as d is larger than certain constant. Overall, we know $\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \lambda_i^2 \mathbb{1}\left\{\mathcal{E}\right\}\right] \geqslant \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \lambda_i^2\right] - 1/3$. Combing with the upper bounds on $\mathbb{E}\left[\bar{\lambda}^2 \mathbb{1}\left\{\mathcal{E}\right\}\right]$, we have $\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n (\lambda_i - \bar{\lambda})^2 \mathbb{1}\left\{\mathcal{E}\right\}\right] \geqslant 1$.

Since conditioning on $\mathcal{E}$, λ_i is bounded by L for all $i \in [n]$. In order to make $\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n (\lambda_i - \bar{\lambda})^2 \mathbb{1}\left\{\mathcal{E}\right\}\right]$ lower bounded by one, there must exist positive constants μ_1, μ_2 such that with probability at least μ_1, $\mathcal{E}$ holds and $\frac{1}{n}\sum_{i=1}^n (\lambda_i - \bar{\lambda})^2 \geqslant \mu_2$.

Since $\frac{1}{n}\sum_{i=1}^n (\lambda_i - \bar{\lambda})^2 \geqslant \mu_2$ and $\lambda_i \leqslant L$ for all $i \in [n]$, we know there exists a subset of eigenvalues $S \subset \{\lambda_i\}_1^n$ with size $\mu_3 n$ such that $|\lambda_i - \bar{\lambda}| \geqslant \mu_4$ for all $\lambda_i \in S$, where μ_3, μ_4 are both positive constants.

If at least half of eigenvalues in S are larger than $\bar{\lambda}$, we know at least $\frac{\mu_3\mu_4 n}{2L}$ number of eigenvalues are smaller than $\bar{\lambda}$. Otherwise, the expectation of the eigenvalues will be larger than $\bar{\lambda}$, which contradicts the definition of $\bar{\lambda}$. Similarly, if at least half of eigenvalues in S are smaller than $\bar{\lambda}$, we know at least $\frac{\mu_3\mu_4 n}{2L}$ number of eigenvalues are larger than $\bar{\lambda}$. Denote $\mu_5 := \frac{\mu_3\mu_4}{2L}$. We know $\lambda_{\mu_5 n} - \lambda_{n-\mu_5 n+1} \geqslant \mu_4$. □

Proof of Lemma D.2.24. Let $\mathcal{E}_1$ be the event that $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\text{train}}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{\text{train}}\| \leqslant \sqrt{d}\sigma$. Let $\mathcal{E}_3$ be the event that $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\text{valid}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\text{valid}}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{\text{valid}}\| \leqslant \sqrt{d}\sigma$. According to Lemma D.2.3 and Lemma D.5.1, we know both $\mathcal{E}_1$ and $\mathcal{E}_3$ hold with probability at least $1 - \exp(-\Omega(d))$. In this proof, we assume both properties hold and take a union bound at the end.

We can lower bound $\|w_{t,\eta} - w_{\text{valid}}\|_{H_{\text{valid}}}^2$ as follows,

$$\begin{aligned}\|w_{t,\eta} - w_{\text{valid}}\|_{H_{\text{valid}}}^2 =& \left\|w_{t,\eta} - w^* - (X_{\text{valid}})^\dagger \xi_{\text{valid}}\right\|_{H_{\text{valid}}}^2 \\ \geqslant& \|w_{t,\eta} - w^*\|_{H_{\text{valid}}}^2 + \frac{1}{n}\|\xi_{\text{valid}}\|^2 - 2\left|\left\langle w_{t,\eta} - w^* \right\rangle H_{\text{valid}} (X_{\text{valid}})^\dagger \xi_{\text{valid}}\right|.\end{aligned}$$

For the second term, by Lemma D.5.1, we know for any $1 > \epsilon > 0$, with probability at least $1 - \exp(-\Omega(\epsilon^2 d))$,

$$\frac{1}{n}\|\xi_{\text{valid}}\|^2 \geqslant (1-\epsilon)\sigma^2.$$

We can write down the third term as $\left\langle [(X_{\text{valid}})^\dagger]^\top H_{\text{valid}}(w_{t,\eta} - w^*) \right\rangle \xi_{\text{valid}}$. Suppose σ is a constant, we know $\left\| [(X_{\text{valid}})^\dagger]^\top H_{\text{valid}}(w_{t,\eta} - w^*) \right\| = O(1/\sqrt{d})$. Therefore, for a fixed $\eta \in [1/L, 3L]$, we have with probability at least $1 - \exp(-\Omega(\epsilon^2 d))$,

$$\left| \left\langle w_{t,\eta} - w^* \right\rangle H_{\text{valid}}(X_{\text{valid}})^\dagger \xi_{\text{valid}} \right| \leqslant \epsilon.$$

To prove this crossing term is small for all $\eta \in [1/L, 3L]$, we need to construct an ϵ-net for the crossing term. Similar as in Lemma D.2.10, we can show there exists an ϵ-net for the crossing term with size $O(t/\epsilon)$. Taking a union bound over this ϵ-net, we are able to show with probability at least $1 - O(t/\epsilon)\exp(-\Omega(\epsilon^2 d))$,

$$\left| \left\langle w_{t,\eta} - w^* \right\rangle H_{\text{valid}}(X_{\text{valid}})^\dagger \xi_{\text{valid}} \right| \leqslant \epsilon,$$

for all $\eta \in [1/L, 3L]$.

Overall, we have with probability at least $1 - O(t/\epsilon)\exp(-\Omega(\epsilon^2 d))$,

$$\begin{aligned}
&\|w_{t,\eta} - w_{\text{valid}}\|_{H_{\text{valid}}}^2 \\
\geqslant &\|w_{t,\eta} - w^*\|_{H_{\text{valid}}}^2 + \frac{1}{n}\|\xi_{\text{valid}}\|^2 - 2\left| \left\langle w_{t,\eta} - w^* \right\rangle H_{\text{valid}}(X_{\text{valid}})^\dagger \xi_{\text{valid}} \right| \\
\geqslant &\|w_{t,\eta} - w^*\|_{H_{\text{valid}}}^2 + (1-\epsilon)\sigma^2 - 2\epsilon \geqslant (1-3\epsilon)\sigma^2,
\end{aligned}$$

for all $\eta \in [1/L, 3L]$, where the last inequality uses $\sigma \geqslant 1$. The proof finishes as we change 3ϵ to ϵ'. $\square$

D.3 Proofs of Train-by-Train with Large Number of Samples (GD)

In this section, we give the proof of Theorem 5.4.2. We show when the size of each training set n and the the number of training tasks m are large enough, train-by-train also performs well. Recall Theorem 5.4.2 as follows.

Theorem 5.4.2. *Let $\widehat{F}_{TbT(n)}(\eta)$ be as defined in Equation (5.3). Assume noise level is a constant c_1. Given any $1 > \epsilon > 0$, assume training set size $n \geqslant \frac{cd}{\epsilon^2}\log\big(\frac{nm}{\epsilon d}\big)$, unroll length $t \geqslant c_2 \log\big(\frac{n}{\epsilon d}\big)$, number of training tasks $m \geqslant \frac{c_3 n^2}{\epsilon^4 d^2}\log\big(\frac{tnm}{\epsilon d}\big)$ and dimension $d \geqslant c_4$ for certain constants c, c_2, c_3, c_4. With probability at least 0.99 in the sampling of training tasks, we have*

$$\mathbb{E}\Big\|w_{t,\eta^*_{train}} - w^*\Big\|^2 \leqslant (1+\epsilon)\frac{d\sigma^2}{n},$$

for all $\eta^*_{train} \in \arg\min_{\eta \geqslant 0} \widehat{F}_{TbT(n)}(\eta)$, where the expectation is taken over new tasks.

In the proof, we use the same notations defined in Section D.2. On each training task P, in Lemma D.3.1 we show the meta-loss can be decomposed into two terms:

$$\Delta_{TbT}(\eta, P) = \frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}} + \frac{1}{2n}\big\|(I_n - \text{Proj}_{X_{\text{train}}})\xi_{\text{train}}\big\|^2,$$

where $w_{\text{train}} = w^* + (X_{\text{train}})^\dagger \xi_{\text{train}}$. Recall that X_{train} is a $n \times d$ matrix with its i-th row as $x_i^\top$. The pseudo-inverse $(X_{\text{train}})^\dagger$ has dimension $d \times n$ satisfying $X^\dagger_{\text{train}} X_{\text{train}} = I_d$. Here, $\text{Proj}_{X_{\text{train}}} \in \mathbb{R}^{n\times n}$ is a projection matrix onto the column span of X_{train}.

In Lemma D.3.1, we show with a constant step size, the first term in $\Delta_{TbT}(\eta, P)$ is exponentially small. The second term is basically the projection of the noise on the orthogonal subspace of the data span. We show this term concentrates well on its mean. This lemma servers as step 1 in Section D.2.1. The proof of Lemma D.3.1 is deferred into Section D.3.1.

Lemma D.3.1. *Assume $n \geqslant 40d$. Given any $1 > \epsilon > 0$, with probability at least $1 - m\exp(-\Omega(n)) - \exp(-\Omega(\epsilon^4 md/n))$,*

$$\widehat{F}_{TbT}(2/3) \leqslant 20(1 - \frac{1}{3})^{2t}\sigma^2 + \frac{n-d}{2n}\sigma^2 + \frac{\epsilon^2 d\sigma^2}{20n}.$$

In the next lemma, we show the empirical meta objective is large when η exceeds certain threshold. We define this threshold $\widehat{\eta}$ such that for any step size larger than $\widehat{\eta}$ the GD sequence has reasonable probability being truncated. In the proof, we rely on the truncated sequences to argue the meta-objective must be high. The precise definition of $\widehat{\eta}$ is in Definition D.3.5. This lemma serves as step 2 in Section D.2.1. We leave the proof of Lemma D.3.2 into Section D.3.2.

Lemma D.3.2. *Let $\widehat{\eta}$ be as defined in Definition D.3.5 with $1 > \epsilon > 0$. Assume $n \geqslant cd, t \geqslant c_2, d \geqslant c_4$ for some constants c, c_2, c_4. With probability at least $1 - \exp(-\Omega(\epsilon^4 md^2/n^2))$,*

$$\widehat{F}_{TbT}(\eta) \geqslant \frac{\epsilon^2 d\sigma^2}{8n} + \frac{n-d}{2n}\sigma^2 - \frac{\epsilon^2 d\sigma^2}{20n},$$

for all $\eta > \widehat{\eta}$.

By Lemma D.3.1 and Lemma D.3.2, we know when t is reasonably large, $\widehat{F}_{TbT}(\eta)$ is larger than $\widehat{F}_{TbT}(2/3)$ for all step sizes $\eta > \widehat{\eta}$. This means the optimal step size $\widehat{\eta}$ must lie in $[0, \widehat{\eta}]$. In Lemma D.3.3, we show a generalization result for $\eta \in [0, \widehat{\eta}]$. This serves as step 3 in Section D.2.1. We prove this lemma in Section D.3.3.

Lemma D.3.3. *Let $\widehat{\eta}$ be as defined in Definition D.3.5 with $1 > \epsilon > 0$. Suppose σ is a constant. Assume $n \geqslant c\log\left(\frac{n}{\epsilon d}\right)d, t \geqslant c_2, d \geqslant c_4$ for some constants c, c_2, c_4. With*

probability at least $1 - m\exp(-\Omega(n)) - O(\frac{tn}{\epsilon^2 d} + m)\exp(-\Omega(m\epsilon^4 d^2/n^2))$,

$$|F_{TbT}(\eta) - \widehat{F}_{TbT}(\eta)| \leqslant \frac{17\epsilon^2 d\sigma^2}{n},$$

for all $\eta \in [0, \widehat{\eta}]$,

Combining Lemma D.3.1, Lemma D.3.2 and Lemma D.3.3, we present the proof of Theorem 5.4.2 as follows.

Proof of Theorem 5.4.2. According to Lemma D.3.1, assuming $n \geqslant 40d$, given any $1/2 > \epsilon > 0$, with probability at least $1 - m\exp(-\Omega(n)) - \exp(-\Omega(\epsilon^4 md/n))$, $\widehat{F}_{TbT}(2/3) \leqslant 20(1 - \frac{1}{3})^{2t}\sigma^2 + \frac{n-d}{2n}\sigma^2 + \frac{\epsilon^2 d\sigma^2}{20n}$. As long as $t \geqslant c_2 \log\left(\frac{n}{\epsilon d}\right)$ for certain constant c_2, we have

$$\widehat{F}_{TbT}(2/3) \leqslant \frac{n-d}{2n}\sigma^2 + \frac{7\epsilon^2 d\sigma^2}{100n}.$$

Let $\widehat{\eta}$ be as defined in Definition D.3.5 with the same ϵ. According to Lemma D.3.2, as long as $n \geqslant cd, t \geqslant c_2, d \geqslant c_4$ with probability at least $1 - \exp(-\Omega(\epsilon^4 md^2/n^2))$,

$$\widehat{F}_{TbT}(\eta) \geqslant \frac{\epsilon^2 d\sigma^2}{8n} + \frac{n-d}{2n}\sigma^2 - \frac{\epsilon^2 d\sigma^2}{20n} = \frac{n-d}{2n}\sigma^2 + \frac{7.5\epsilon^2 d\sigma^2}{100n}$$

for all $\eta > \widehat{\eta}$. We have $\widehat{F}_{TbT}(\eta) > \widehat{F}_{TbT}(2/3)$ for all $\eta \geqslant \widehat{\eta}$. This implies that η^*_{train} is within $[0, \widehat{\eta}]$ and $\widehat{F}_{TbT}(\eta^*_{\text{train}}) \leqslant \widehat{F}_{TbT}(2/3) \leqslant \frac{n-d}{2n}\sigma^2 + \frac{7\epsilon^2 d\sigma^2}{100n}$.

By Lemma D.3.3, assuming σ is a constant and assuming $n \geqslant c\log\left(\frac{n}{\epsilon d}\right)d$ for some constant c, we have with probability at least $1 - m\exp(-\Omega(n)) - O(\frac{tn}{\epsilon^2 d} + m)\exp(-\Omega(m\epsilon^4 d^2/n^2))$,

$$|F_{TbT}(\eta) - \widehat{F}_{TbT}(\eta)| \leqslant \frac{17\epsilon^2 d\sigma^2}{n},$$

for all $\eta \in [0, \widehat{\eta}]$. This then implies

$$F_{T\not bT}(\eta^*_{\text{train}}) \leqslant \widehat{F}_{T\not bT}(\eta^*_{\text{train}}) + \frac{17\epsilon^2 d\sigma^2}{n} \leqslant \frac{n-d}{2n}\sigma^2 + \frac{24\epsilon^2 d\sigma^2}{n}.$$

By the analysis in Lemma D.3.1, we have

$$\begin{aligned} F_{T\not bT}(\eta^*_{\text{train}}) =& \mathbb{E}\frac{1}{2}\left\|w_{t,\eta^*_{\text{train}}} - w_{\text{train}}\right\|^2_{H_{\text{train}}} + \mathbb{E}\frac{1}{2n}\left\|(I_n - \text{Proj}_{X_{\text{train}}})\xi_{\text{train}}\right\|^2 \\ =& \mathbb{E}\frac{1}{2}\left\|w_{t,\eta^*_{\text{train}}} - w_{\text{train}}\right\|^2_{H_{\text{train}}} + \frac{n-d}{2n}\sigma^2. \end{aligned}$$

Therefore, we know $\mathbb{E}\frac{1}{2}\left\|w_{t,\eta^*_{\text{train}}} - w_{\text{train}}\right\|^2_{H_{\text{train}}} \leqslant \frac{24\epsilon^2 d\sigma^2}{n}$. Next, we show this implies $\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2$ is small.

Let $\mathcal{E}$ be the event that $1-\epsilon \leqslant \lambda_i(H_{\text{train}}) \leqslant 1+\epsilon$ for all $i \in [d]$. According to Lemma D.3.4, we know $\Pr[\mathcal{E}] \geqslant 1 - \exp(-\Omega(\epsilon^2 n))$ as long as $n \geqslant 10d/\epsilon^2$. Then, we can decompose $\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2$ as follows,

$$\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2 = \mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2 \mathbb{1}\left\{\mathcal{E}\right\} + \mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2 \mathbb{1}\left\{\bar{\mathcal{E}}\right\}.$$

Let's first show the second term is small. Due to the truncation in our algorithm, we know $\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2 \leqslant 41^2\sigma^2$, which then implies $\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2 \mathbb{1}\left\{\bar{\mathcal{E}}\right\} \leqslant 41^2\sigma^2 \exp(-\Omega(\epsilon^2 n))$. As long as $n \geqslant \frac{c}{\epsilon^2}\log\left(\frac{n}{\epsilon d}\right)$ for some constant c, we will have that $\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2 \mathbb{1}\left\{\bar{\mathcal{E}}\right\} \leqslant \frac{\epsilon d\sigma^2}{n}$.

We can upper bound the first term by Young's inequality,

$$\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2 \mathbb{1}\left\{\mathcal{E}\right\} \leqslant (1+\frac{1}{\epsilon})\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w_{\text{train}}\right\|^2 \mathbb{1}\left\{\mathcal{E}\right\} + (1+\epsilon)\mathbb{E}\|w_{\text{train}} - w^*\|^2 \mathbb{1}\left\{\mathcal{E}\right\}.$$

Conditioning on $\mathcal{E}$, we have $\left\|w_{t,\eta^*_{\text{train}}} - w_{\text{train}}\right\|^2_{H_{\text{train}}} \geqslant (1-\epsilon)\left\|w_{t,\eta^*_{\text{train}}} - w_{\text{train}}\right\|^2$ which

implies $\left\|w_{t,\eta^*_{\text{train}}} - w_{\text{train}}\right\|^2 \leqslant (1+2\epsilon)\left\|w_{t,\eta^*_{\text{train}}} - w_{\text{train}}\right\|^2_{H_{\text{train}}}$ as long as $\epsilon \leqslant 1/2$. Similarly, we also have $\|w_{\text{train}} - w^*\|^2 \leqslant (1+2\epsilon)\|w_{\text{train}} - w^*\|^2_{H_{\text{train}}}$. Then, we have

$$\begin{aligned}
&\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2 \mathbb{1}\{\mathcal{E}\} \\
\leqslant&(1+\frac{1}{\epsilon})(1+2\epsilon)\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w_{\text{train}}\right\|^2_{H_{\text{train}}} \mathbb{1}\{\mathcal{E}\} \\
&+ (1+\epsilon)(1+2\epsilon)\mathbb{E}\|w_{\text{train}} - w^*\|^2_{H_{\text{train}}} \mathbb{1}\{\mathcal{E}\} \\
\leqslant&(5+\frac{1}{\epsilon})\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w_{\text{train}}\right\|^2_{H_{\text{train}}} + (1+5\epsilon)\mathbb{E}\|w_{\text{train}} - w^*\|^2_{H_{\text{train}}} \\
\leqslant&(5+\frac{1}{\epsilon})\frac{48\epsilon^2 d\sigma^2}{n} + (1+5\epsilon)\frac{d\sigma^2}{n} \leqslant (1+293\epsilon)\frac{d\sigma^2}{n}.
\end{aligned}$$

Overall, we have $\mathbb{E}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2 \leqslant (1+293\epsilon)\frac{d\sigma^2}{n} + \frac{\epsilon d\sigma^2}{n} = (1+294\epsilon)\frac{d\sigma^2}{n}$. Combining all the conditions, we know this holds with probability at least 0.99 as long as σ is a constant c_1, $n \geqslant \frac{cd}{\epsilon^2}\log\left(\frac{nm}{\epsilon d}\right), t \geqslant c_2\log\left(\frac{n}{\epsilon d}\right), m \geqslant \frac{c_3 n^2}{\epsilon^4 d^2}\log\left(\frac{tnm}{\epsilon d}\right), d \geqslant c_4$ for some constants c, c_2, c_3, c_4. We finish the proof by choosing $\epsilon = \epsilon'/294$. □

D.3.1 Upper Bounding $\widehat{F}_{TbT}(2/3)$

In this section, we show there exists a step size that achieves small empirical meta objective. On each training task P, we show the meta-loss can be decomposed into two terms:

$$\begin{aligned}
\Delta_{TbT}(\eta, P) =&\frac{1}{2n}\sum_{i=1}^{n}\left(\langle w_{t,\eta} - w_{\text{train}}\rangle x_i - \left(\xi_i - x_i^\top X^\dagger_{\text{train}}\xi_{\text{train}}\right)\right)^2 \\
=&\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}} + \frac{1}{2n}\left\|(I_n - \text{Proj}_{X_{\text{train}}})\xi_{\text{train}}\right\|^2,
\end{aligned}$$

where $w_{\text{train}} = w^* + (X_{\text{train}})^\dagger\xi_{\text{train}}$. In Lemma D.3.1, we show with a constant step size, the first term is exponentially small and the second term concentrates on its

mean.

Lemma D.3.1. *Assume $n \geqslant 40d$. Given any $1 > \epsilon > 0$, with probability at least $1 - m\exp(-\Omega(n)) - \exp(-\Omega(\epsilon^4 md/n))$,*

$$\widehat{F}_{TbT}(2/3) \leqslant 20(1 - \frac{1}{3})^{2t}\sigma^2 + \frac{n-d}{2n}\sigma^2 + \frac{\epsilon^2 d\sigma^2}{20n}.$$

Before we go to the proof of Lemma D.3.1, let's first show the covariance matrix H_{train} is very close to identity when n is much larger than d. The proof follows from the concentration of singular values of random Gaussian matrix (Lemma D.5.4). We leave the proof into Section D.3.4.

Lemma D.3.4. *Given $1 > \epsilon > 0$, assume $n \geqslant 10d/\epsilon^2$. With probability at least $1 - \exp(-\Omega(\epsilon^2 n))$,*

$$(1-\epsilon)\sqrt{n} \leqslant \sigma_i(X_{train}) \leqslant (1+\epsilon)\sqrt{n} \text{ and } 1 - \epsilon \leqslant \lambda_i(H_{train}) \leqslant 1 + \epsilon,$$

for all $i \in [d]$.

Now, we are ready to present the proof of Lemma D.3.1.

Proof of Lemma D.3.1. Let's first look at one training set S_{train}, in which $y_i = \langle w^* \rangle x_i + \xi_i$ for each sample. Recall the meta-loss as

$$\Delta_{TbT}(\eta, P) = \frac{1}{2n}\sum_{i=1}^{n}\left(\langle w_{t,\eta}\rangle x_i - \langle w^*\rangle x_i - \xi_i\right)^2.$$

Recall that X_{train} is an $n \times d$ matrix with its i-th row as $x_i^\top$. With probability 1, we know X_{train} is full column rank. Denote the pseudo-inverse of X_{train} as $X_{\text{train}}^\dagger \in \mathbb{R}^{d\times n}$ that satisfies $X_{\text{train}}^\dagger X_{\text{train}} = I_d$ and $X_{\text{train}}X_{\text{train}}^\dagger = \text{Proj}_{X_{\text{train}}}$, where $\text{Proj}_{X_{\text{train}}} \in \mathbb{R}^{n\times n}$ is a projection matrix onto the column span of X_{train}.

Let w_{train} be $w^* + X_{\text{train}}^{\dagger}\xi_{\text{train}}$, where ξ_{train} is an n-dimensional vector with its i-th entry as ξ_i. We have,

$$\begin{aligned}
&\Delta_{TbT}(\eta, P) \\
=&\frac{1}{2n}\sum_{i=1}^{n}\left(\langle w_{t,\eta} - w_{\text{train}}\rangle x_i - \left(\xi_i - x_i^{\top} X_{\text{train}}^{\dagger}\xi_{\text{train}}\right)\right)^2 \\
=&\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 + \frac{1}{2n}\left\|(I_n - \text{Proj}_{X_{\text{train}}})\xi_{\text{train}}\right\|^2 \\
&- \frac{1}{n}\sum_{i=1}^{n}\langle w_{t,\eta} - w_{\text{train}}\rangle x_i\xi_i - x_i x_i^{\top} X_{\text{train}}^{\dagger}\xi_{\text{train}}.
\end{aligned}$$

We first show the crossing term is actually zero. We have,

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\langle w_{t,\eta} - w_{\text{train}}\rangle x_i\xi_i - x_i x_i^{\top} X_{\text{train}}^{\dagger}\xi_{\text{train}} \\
=&\frac{1}{n}\langle w_{t,\eta} - w_{\text{train}}\rangle \sum_{i=1}^{n} x_i\xi_i - \sum_{i=1}^{n} x_i x_i^{\top} X_{\text{train}}^{\dagger}\xi_{\text{train}} \\
=&\frac{1}{n}\langle w_{t,\eta} - w_{\text{train}}\rangle X_{\text{train}}^{\top}\xi_{\text{train}} - X_{\text{train}}^{\top} X_{\text{train}} X_{\text{train}}^{\dagger}\xi_{\text{train}} \\
=&\frac{1}{n}\langle w_{t,\eta} - w_{\text{train}}\rangle X_{\text{train}}^{\top}\xi_{\text{train}} - X_{\text{train}}^{\top}\xi_{\text{train}} = 0,
\end{aligned}$$

where the second last equality holds because $X_{\text{train}}X_{\text{train}}^{\dagger} = \text{Proj}_{X_{\text{train}}}$.

We can define $w_{\text{train}}^{(k)}$ as $w_k^* + (X_{\text{train}}^{(k)})^{\dagger}\xi_{\text{train}}^{(k)}$ for every training set $S_{\text{train}}^{(k)}$. Then, we have

$$\widehat{F}_{TbT}(\eta) = \frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|_{H_{\text{train}}^{(k)}}^2 + \frac{1}{m}\sum_{k=1}^{m}\frac{1}{2n}\left\|(I_n - \text{Proj}_{X_{\text{train}}^{(k)}})\xi_{\text{train}}^{(k)}\right\|^2$$

We first prove that the second term concentrates on its mean. We can concatenate m noise vectors $\xi_{\text{train}}^{(k)}$ into a single noise vector $\bar{\xi}_{\text{train}}$ with dimension nm. We can also construct a data matrix $\bar{X}_{\text{train}} \in \mathbb{R}^{nm\times dm}$ that consists of $X_{\text{train}}^{(k)}$ as diagonal blocks.

Then the second term can be written as

$$\frac{1}{2}\left\|\frac{1}{\sqrt{nm}}(I_{nm}-\mathrm{Proj}_{\bar{X}_{\mathrm{train}}})\bar{\xi}_{\mathrm{train}}\right\|^2.$$

According to Lemma D.5.1, with probability at least $1-\exp(-\Omega(\epsilon^4 md^2/n))$,

$$\left(1-\frac{\epsilon^2 d}{n}\right)\sigma \leqslant \frac{1}{\sqrt{nm}}\left\|\bar{\xi}_{\mathrm{train}}\right\| \leqslant \left(1+\frac{\epsilon^2 d}{n}\right)\sigma.$$

By Johnson-Lindenstrauss Lemma (Lemma D.5.5), we know with probability at least $1-\exp(-\Omega(\epsilon^4 md))$,

$$\frac{1}{\sqrt{nm}}\left\|\mathrm{Proj}_{\bar{X}_{\mathrm{train}}}\bar{\xi}_{\mathrm{train}}\right\| \geqslant (1-\epsilon^2)\frac{\sqrt{md}}{\sqrt{mn}}\frac{1}{\sqrt{nm}}\left\|\bar{\xi}_{\mathrm{train}}\right\| \geqslant (1-\epsilon^2)\sqrt{\frac{d}{n}}(1-\frac{\epsilon^2 d}{n})\sigma.$$

Therefore, we have $\left\|\frac{1}{\sqrt{nm}}\bar{\xi}_{\mathrm{train}}\right\|^2 \leqslant (1+\frac{3\epsilon^2 d}{n})\sigma^2$ and $\left\|\frac{1}{\sqrt{nm}}\mathrm{Proj}_{\bar{X}_{\mathrm{train}}}\bar{\xi}_{\mathrm{train}}\right\|^2 \geqslant (1-2\epsilon^2)\frac{d}{n}\sigma^2$. Overall, we know with probability at least $1-\exp(-\Omega(\epsilon^4 md/n))$,

$$\frac{1}{2}\left\|\frac{1}{\sqrt{nm}}(I_{nm}-\mathrm{Proj}_{\bar{X}_{\mathrm{train}}})\bar{\xi}_{\mathrm{train}}\right\|^2 \leqslant \frac{n-d}{2n}\sigma^2+\frac{5\epsilon^2 d\sigma^2}{2n}.$$

Now, we show the first term in meta objective is small when we choose a right step size. According to Lemma D.3.4, we know as long as $n \geqslant 40d$, with probability at least $1-\exp(-\Omega(n))$, $\sqrt{n}/2 \leqslant \sigma_i(X_{\mathrm{train}}^{(k)}) \leqslant 3\sqrt{n}/2$ and $1/2 \leqslant \lambda_i(H_{\mathrm{train}}^{(k)}) \leqslant 3/2$, for all $i \in [d]$. According to Lemma D.5.1, we know with probability at least $1-\exp(-\Omega(n))$, $\left\|\xi_{\mathrm{train}}^{(k)}\right\| \leqslant 2\sqrt{n}\sigma$. Taking a union bound on m tasks, we know all these events hold with probability at least $1-m\exp(-\Omega(n))$.

For each $k \in [m]$, we have $\left\|w_{\mathrm{train}}^{(k)}\right\| \leqslant 1+\frac{2}{\sqrt{n}}2\sqrt{n}\sigma \leqslant 5\sigma$. It's easy to verify that for any step size at most $2/3$, the GD sequence will not be truncated since we choose

the threshold norm as 40σ. Then, for any step size $\eta \leqslant 2/3$, we have

$$\begin{aligned}\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)}-w_{\text{train}}^{(k)}\right\|_{H_{\text{train}}^{(k)}}^{2} =&\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|(I-\eta H_{\text{train}}^{(k)})^{t}w_{\text{train}}^{(k)}\right\|_{H_{\text{train}}^{(k)}}^{2}\\ \leqslant&\frac{3}{4}(1-\frac{\eta}{2})^{2t}25\sigma^{2}\leqslant 20(1-\frac{1}{3})^{2t}\sigma^{2},\end{aligned}$$

where the last inequality chooses η as $2/3$.

Overall, we know with probability at least $1-m\exp(-\Omega(n))-\exp(-\Omega(\epsilon^{4}md/n))$,

$$\widehat{F}_{TbT}(2/3)\leqslant 20(1-\frac{1}{3})^{2t}\sigma^{2}+\frac{n-d}{2n}\sigma^{2}+\frac{5\epsilon^{2}d\sigma^{2}}{2n}.$$

We finish the proof by changing $\frac{5\epsilon^2}{2}$ by $(\epsilon')^2/20$. □

D.3.2 Lower Bounding $\widehat{F}_{TbT}$ for $\eta \in (\widehat{\eta}, \infty)$

In this section, we show the empirical meta objective is large when the step size exceeds certain threshold. Recall Lemma D.3.2 as follows.

Lemma D.3.2. *Let $\widehat{\eta}$ be as defined in Definition D.3.5 with $1 > \epsilon > 0$. Assume $n \geqslant cd, t \geqslant c_2, d \geqslant c_4$ for some constants c, c_2, c_4. With probability at least $1 - \exp(-\Omega(\epsilon^4 md^2/n^2))$,*

$$\widehat{F}_{TbT}(\eta)\geqslant\frac{\epsilon^{2}d\sigma^{2}}{8n}+\frac{n-d}{2n}\sigma^{2}-\frac{\epsilon^{2}d\sigma^{2}}{20n},$$

for all $\eta > \widehat{\eta}$.

Roughly speaking, we define $\widehat{\eta}$ such that for any step size larger than $\widehat{\eta}$ the GD sequence has a reasonable probability being truncated. The definition is very similar as $\widetilde{\eta}$ in Definition D.2.5.

Definition D.3.5. Given a training task P, let $\mathcal{E}_1$ be the event that $\sqrt{n}/2 \leqslant \sigma_i(X_{\text{train}}) \leqslant 3\sqrt{n}/2$ and $1/2 \leqslant \lambda_i(H_{\text{train}}) \leqslant 3/2$ for all $i \in [d]$ and $\sqrt{n}\sigma/2 \leqslant \|\xi_{\text{train}}\| \leqslant 2\sqrt{n}\sigma$. Let $\bar{\mathcal{E}}_2(\eta)$ be the event that the GD sequence is truncated with step size η. Given $1 > \epsilon > 0$, define $\widehat{\eta}$ as follows,

$$\widehat{\eta} = \inf\left\{\eta \geqslant 0 \middle| \mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\} \geqslant \frac{\epsilon^2 d\sigma^2}{n}\right\}.$$

Similar as in Lemma D.2.6, we show $\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta')\right\} \geqslant \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\}$ for any $\eta' \geqslant \eta$. This means conditioning on $\mathcal{E}_1$, if a GD sequence gets truncated with step size η, it has to be truncated with any step size $\eta' \geqslant \eta$. The proof is deferred into Section D.3.4.

Lemma D.3.6. *Fixing a training set S_{train}, let $\mathcal{E}_1$ and $\bar{\mathcal{E}}_2(\eta)$ be as defined in Definition D.3.5. We have*

$$\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta')\right\} \geqslant \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\},$$

for any $\eta' \geqslant \eta$.

Next, we show $\widehat{\eta}$ does exist and is a constant. Similar as in Lemma D.2.7, we show that the GD sequence almost never diverges when η is small and diverges with high probability when η is large. The proof is left in Section D.3.4.

Lemma D.3.7. *Let $\widehat{\eta}$ be as defined in Definition D.3.5. Suppose σ is a constant. Assume $n \geqslant cd, t \geqslant c_2, d \geqslant c_4$ for some constants c, c_2, c_4. We have*

$$\frac{4}{3} < \widetilde{\eta} < 6.$$

Next, we show the empirical loss is large for any η larger than $\widetilde{\eta}$. The proof is very similar as the proof of Lemma 5.4.4.

Proof of Lemma D.3.2. By Lemma D.3.7, we know $\widehat{\eta}$ is a constant as long as $n \geqslant cd, t \geqslant c_2, d \geqslant c_4$ for some constants c, c_2, c_4. Let $\mathcal{E}_1$ and $\bar{\mathcal{E}}_2(\eta)$ be as defined in Definition D.3.5. For simplicity of proof, we assume $\mathbb{E}\frac{1}{2}\|w_{t,\widehat{\eta}} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widehat{\eta})\right\} \geqslant \frac{\epsilon^2 d\sigma^2}{n}$. The other case can be resolved using same techniques in Lemma 5.4.4

Conditioning on $\mathcal{E}_1$, we know $\frac{1}{2}\|w_{t,\widehat{\eta}} - w_{\text{train}}\|^2_{H_{\text{train}}} \leqslant \frac{3}{4}45^2\sigma^2$. Therefore, we know $\Pr\big[\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widehat{\eta})\big] \geqslant \frac{4\epsilon^2 d}{3\times 45^2 n}$. For each task k, define $\mathcal{E}_1^{(k)}$ and $\bar{\mathcal{E}}_2^{(k)}(\eta)$ as the corresponding events on training set $S^{(k)}_{\text{train}}$. By Hoeffding's inequality, we know with probability at least $1 - \exp(-\Omega(\epsilon^4 m d^2/n^2))$,

$$\frac{1}{m}\sum_{k=1}^{m} \mathbb{1}\left\{\mathcal{E}_1^{(k)} \cap \bar{\mathcal{E}}_2^{(k)}(\widehat{\eta})\right\} \geqslant \frac{\epsilon^2 d}{45^2 n}.$$

By Lemma D.3.6, we know $\mathbb{1}\left\{\mathcal{E}_1^{(k)} \cap \bar{\mathcal{E}}_2^{(k)}(\eta)\right\} \geqslant \mathbb{1}\left\{\mathcal{E}_1^{(k)} \cap \bar{\mathcal{E}}_2^{(k)}(\widehat{\eta})\right\}$ for any $\eta \geqslant \widehat{\eta}$.

Recall that

$$\widehat{F}_{TbT}(\eta) = \frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2_{H^{(k)}_{\text{train}}} + \frac{1}{m}\sum_{k=1}^{m}\frac{1}{2n}\left\|(I_n - \text{Proj}_{X^{(k)}_{\text{train}}})\xi^{(k)}_{\text{train}}\right\|^2.$$

We can lower bound the first term for any $\eta > \widehat{\eta}$ as follows,

$$\begin{aligned}
\widehat{F}_{TbT}(\eta) &= \frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2_{H^{(k)}_{\text{train}}} \\
&\geqslant \frac{1}{m}\sum_{k=1}^{m}\frac{1}{2}\left\|w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2_{H^{(k)}_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1^{(k)} \cap \bar{\mathcal{E}}_2^{(k)}(\eta)\right\} \\
&\geqslant \frac{35^2\sigma^2}{4}\frac{1}{m}\sum_{k=1}^{m}\mathbb{1}\left\{\mathcal{E}_1^{(k)} \cap \bar{\mathcal{E}}_2^{(k)}(\eta)\right\} \\
&\geqslant \frac{35^2\sigma^2}{4}\frac{1}{m}\sum_{k=1}^{m}\mathbb{1}\left\{\mathcal{E}_1^{(k)} \cap \bar{\mathcal{E}}_2^{(k)}(\widehat{\eta})\right\} \geqslant \frac{\epsilon^2 d\sigma^2}{8n},
\end{aligned}$$

where the second inequality lower bounds the loss for one task by $35^2\sigma^2$ when the

sequence gets truncated.

For the second term, according to the analysis in Lemma D.3.1, with probability at least $1 - \exp(-\Omega(\epsilon^4 md/n))$,

$$\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2n}\left\|(I_n - \mathrm{Proj}_{X_{\mathrm{train}}^{(k)}})\xi_{\mathrm{train}}^{(k)}\right\|^2 \geqslant \frac{n-d}{2n}\sigma^2 - \frac{\epsilon^2 d\sigma^2}{20n}.$$

Overall, with probability at least $1 - \exp(-\Omega(\epsilon^4 md^2/n^2))$,

$$\widehat{F}_{TbT}(\eta) \geqslant \frac{\epsilon^2 d\sigma^2}{8n} + \frac{n-d}{2n}\sigma^2 - \frac{\epsilon^2 d\sigma^2}{20n},$$

for all $\eta > \widehat{\eta}$. □

D.3.3 Generalization for $\eta \in [0, \widehat{\eta}]$

Combing Lemma D.3.1 and Lemma D.3.2, it's not hard to see that the optimal step size η^*_{train} lies in $[0, \widehat{\eta}]$. In this section, we show a generalization result for step sizes in $[0, \widehat{\eta}]$. The proof of Lemma D.3.3 is given at the end of this section.

Lemma D.3.3. *Let $\widehat{\eta}$ be as defined in Definition D.3.5 with $1 > \epsilon > 0$. Suppose σ is a constant. Assume $n \geqslant c\log\left(\frac{n}{\epsilon d}\right)d, t \geqslant c_2, d \geqslant c_4$ for some constants c, c_2, c_4. With probability at least $1 - m\exp(-\Omega(n)) - O(\frac{tn}{\epsilon^2 d} + m)\exp(-\Omega(m\epsilon^4 d^2/n^2))$,*

$$|F_{TbT}(\eta) - \widehat{F}_{TbT}(\eta)| \leqslant \frac{17\epsilon^2 d\sigma^2}{n},$$

for all $\eta \in [0, \widehat{\eta}]$,

In Lemma D.3.8, we show $\widehat{F}_{TbT}$ concentrates on F_{TbT} at any fixed step size. The proof is almost the same as Lemma D.2.8. We omit its proof.

Lemma D.3.8. *Suppose σ is a constant. For any fixed η and any $1 > \epsilon > 0$, with probability at least $1 - \exp(-\Omega(\epsilon^2 m))$,*

$$\left|\widehat{F}_{TbT}(\eta) - F_{TbT}(\eta)\right| \leqslant \epsilon.$$

Next, we construct an ϵ-net for F_{TbT} in $[0, \widehat{\eta}]$. The proof is very similar as in Lemma D.2.9. We defer its proof into Section D.3.4.

Lemma D.3.9. *Let $\widehat{\eta}$ be as defined in Definition D.3.5 with $1 > \epsilon > 0$. Assume the conditions in Lemma D.3.7 hold. Assume $n \geqslant c \log\left(\frac{n}{\epsilon d}\right) d$ for some constant c. There exists an $\frac{8\epsilon^2 d\sigma^2}{n}$-net $N \subset [0, \widehat{\eta}]$ for F_{TbT} with $|N| = O(\frac{tn}{\epsilon^2 d})$. That means, for any $\eta \in [0, \widehat{\eta}]$,*

$$|F_{TbT}(\eta) - F_{TbT}(\eta')| \leqslant \frac{8\epsilon^2 d\sigma^2}{n},$$

for $\eta' = \arg\min_{\eta'' \in N, \eta'' \leqslant \eta}(\eta - \eta'')$.

We also construct an ϵ-net for the empirical meta objective. The proof is very similar as in Lemma D.2.10. We leave its proof into Section D.3.4.

Lemma D.3.10. *Let $\widehat{\eta}$ be as defined in Definition D.3.5 with $1 > \epsilon > 0$. Assume the conditions in Lemma D.3.7 hold. Assume $n \geqslant 40d$. With probability at least $1 - m\exp(-\Omega(n))$, there exists an $\frac{\epsilon^2 d\sigma^2}{n}$-net $N' \subset [0, \widehat{\eta}]$ for $\widehat{F}_{TbT}$ with $|N'| = O(\frac{tn}{\epsilon^2 d} + m)$. That means, for any $\eta \in [0, \widehat{\eta}]$,*

$$|\widehat{F}_{TbT}(\eta) - \widehat{F}_{TbT}(\eta')| \leqslant \frac{\epsilon^2 d\sigma^2}{n},$$

for $\eta' = \arg\min_{\eta'' \in N', \eta'' \leqslant \eta}(\eta - \eta'')$.

Combing the above three lemmas, we give the proof of Lemma D.3.3.

Proof of Lemma D.3.3. We assume σ as a constant in this proof. By Lemma D.3.8, we know with probability at least $1 - \exp(-\Omega(m\epsilon^4 d^2/n^2))$, $\left|\widehat{F}_{TbT}(\eta) - F_{TbT}(\eta)\right| \leqslant \frac{\epsilon^2 d\sigma^2}{n}$ for any fixed η. By Lemma D.3.9, we know as long as $n \geqslant c\log\left(\frac{n}{\epsilon d}\right)d$ for some constant c, there exists an $\frac{8\epsilon^2 d\sigma^2}{n}$-net N for F_{TbT} with size $O(\frac{tn}{\epsilon^2 d})$. By Lemma D.3.10, we know with probability at least $1 - m\exp(-\Omega(n))$, there exists an $\frac{\epsilon^2 d\sigma^2}{n}$-net N' for $\widehat{F}_{TbT}$ with size $O(\frac{tn}{\epsilon^2 d} + m)$. It's not hard to verify that $N \cup N'$ is still an $\frac{8\epsilon^2 d\sigma^2}{n}$-net for $\widehat{F}_{TbV}$ and F_{TbV}. That means, for any $\eta \in [0, \widehat{\eta}]$, we have

$$|F_{TbT}(\eta) - F_{TbT}(\eta')|, |\widehat{F}_{TbT}(\eta) - \widehat{F}_{TbT}(\eta')| \leqslant \frac{8\epsilon^2 d\sigma^2}{n},$$

for $\eta' = \arg\min_{\eta'' \in N \cup N', \eta'' \leqslant \eta}(\eta - \eta'')$.

Taking a union bound over $N \cup N'$, we have with probability at least $1 - O(\frac{tn}{\epsilon^2 d} + m)\exp(-\Omega(m\epsilon^4 d^2/n^2))$,

$$\left|\widehat{F}_{TbT}(\eta) - F_{TbT}(\eta)\right| \leqslant \frac{\epsilon^2 d\sigma^2}{n}$$

for all $\eta \in N \cup N'$.

Overall, with probability at least $1 - m\exp(-\Omega(n)) - O(\frac{tn}{\epsilon^2 d} + m)\exp(-\Omega(m\epsilon^4 d^2/n^2))$, for all $\eta \in [0, \widehat{\eta}]$,

$$\begin{aligned}
&|F_{TbT}(\eta) - \widehat{F}_{TbT}(\eta)| \\
\leqslant &|F_{TbT}(\eta) - F_{TbT}(\eta')| + |\widehat{F}_{TbT}(\eta) - \widehat{F}_{TbT}(\eta')| + |\widehat{F}_{TbT}(\eta') - F_{TbT}(\eta')| \\
\leqslant &\frac{17\epsilon^2 d\sigma^2}{n},
\end{aligned}$$

where $\eta' = \arg\min_{\eta'' \in N \cup N', \eta'' \leqslant \eta}(\eta - \eta'')$. □

Proof of Lemma D.3.4. According to Lemma D.5.4, we know with probability at least $1 - 2\exp(-t^2/2)$,

$$\sqrt{n} - \sqrt{d} - t \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{n} + \sqrt{d} + t$$

for all $i \in [d]$. Since $d \leqslant \frac{\epsilon^2 n}{10}$, we have $\sqrt{n} - \frac{\epsilon\sqrt{n}}{\sqrt{10}} - t \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{n} + \frac{\epsilon\sqrt{n}}{\sqrt{10}} + t$. Choosing $t = (\frac{1}{3} - \frac{1}{\sqrt{10}})\epsilon\sqrt{n}$, we have with probability at least $1 - \exp(-\Omega(\epsilon^2 n))$,

$$(1 - \frac{\epsilon}{3})\sqrt{n} \leqslant \sigma_i(X_{\text{train}}) \leqslant (1 + \frac{\epsilon}{3})\sqrt{n}.$$

Since $\lambda_i(H_{\text{train}}) = 1/n\sigma_i^2(X_{\text{train}})$, we have $1 - \epsilon \leqslant \lambda_i(H_{\text{train}}) \leqslant 1 + \epsilon$. □

Proof of Lemma D.3.6. The proof is almost the same as in Lemma D.2.6. We omit the details here. Basically, in Lemma D.2.6, the only property we rely on is that the norm threshold is larger than $2\|w_{\text{train}}\|$ conditioning on $\mathcal{E}_1$. Conditioning on $\mathcal{E}_1$, we know $\|w_{\text{train}}\| \leqslant 5\sigma$. Recall that the norm threshold is still set as 40σ. So this property is preserved and the previous proof works. □

Proof of Lemma D.3.7. The proof is very similar as in Lemma D.2.7. Conditioning on $\mathcal{E}_1$, we know $\|H_{\text{train}}\| \leqslant 3/2$ and $\|w_{\text{train}}\| \leqslant 5\sigma$. So the GD sequence never exceeds the norm threshold 40σ for any $\eta \leqslant 4/3$. That means,

$$\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\} = 0$$

for all $\eta \leqslant 4/3$.

To lower bound the loss for large step size, we need to first lower bound $\|w_{\text{train}}\|$. Recall that $w_{\text{train}} = w^* + (X_{\text{train}})^\dagger \xi_{\text{train}}$. Conditioning on $\mathcal{E}_1$, we know $\|\xi_{\text{train}}\| \leqslant 2\sqrt{n}\sigma$ and $\sigma_d(X_{\text{train}}) \geqslant \sqrt{n}/2$, which implies $\left\|(X_{\text{train}})^\dagger\right\| \leqslant 2/\sqrt{n}$. By Johnson-Lindenstrauss

Lemma (Lemma D.5.5), we have $\left\|\mathrm{Proj}_{X_{\mathrm{train}}}\xi_{\mathrm{train}}\right\| \leqslant \frac{3}{2}\sqrt{d/n}\left\|\xi_{\mathrm{train}}\right\|$ with probability at least $1-\exp(-\Omega(d))$. Call this event $\mathcal{E}_3$. Conditioning on $\mathcal{E}_1 \cap \mathcal{E}_3$, we have

$$\left\|(X_{\mathrm{train}})^{\dagger}\xi_{\mathrm{train}}\right\| \leqslant 2\sqrt{n}\sigma\frac{2}{\sqrt{n}}\frac{3}{2}\sqrt{\frac{d}{n}} \leqslant 6\sqrt{\frac{d}{n}}\sigma,$$

which is smaller than $1/2$ as long as $n \geqslant 12^2 d\sigma^2$. Note that we assume σ is a constant. This then implies $\|w_{\mathrm{train}}\| \geqslant 1/2$.

Let $\{w'_{\tau,\eta}\}$ be the GD sequence without truncation. For any step size $\eta \in [6,\infty]$, conditioning on $\mathcal{E}_1 \cap \mathcal{E}_3$, we have

$$\left\|w'_{t,\eta}\right\| \geqslant \left((6\times\frac{1}{2}-1)^t - 1\right)\|w_{\mathrm{train}}\| \geqslant \left(2^t-1\right)\frac{1}{2} \geqslant 40\sigma,$$

where the last inequality holds as long as $t \geqslant c_2$ for some constant c_2. Therefore, we know when $\eta \in [6,\infty)$, $\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\} = \mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_3\right\}$. Assuming $n \geqslant 40d$, we know $\mathcal{E}_1$ holds with probability at least $1-\exp(-\Omega(n))$. Then, we have for any $\eta \geqslant 6$,

$$\mathbb{E}\frac{1}{2}\|w_{t,\eta}-w_{\mathrm{train}}\|^2_{H_{\mathrm{train}}}\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\} \geqslant \frac{1}{4}\left(40\sigma-5\sigma\right)^2\Pr[\mathcal{E}_1\cap\mathcal{E}_3] \geqslant \frac{\epsilon^2 d\sigma^2}{n},$$

where the last inequality assumes $n \geqslant c, d \geqslant c_4$ for some constant c, c_4.

Overall, we know $\mathbb{E}\frac{1}{2}\|w_{t,\eta}-w_{\mathrm{train}}\|^2_{H_{\mathrm{train}}}\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\eta)\right\}$ equals zero for all $\eta \in [0,4/3]$ and is at least $\frac{\epsilon^2 d\sigma^2}{n}$ for all $\eta \in [6,\infty)$. By definition, we know $\widehat{\eta} \in (4/3, 6)$. $\square$

Proof of Lemma D.3.9. By Lemma D.3.7, we know $\widehat{\eta}$ is a constant. The proof is very similar as in Lemma D.2.9. Let $\mathcal{E}_1$ and $\bar{\mathcal{E}}_2(\eta)$ be as defined in Definition D.3.5. For the simplicity of the proof, we assume $\mathbb{E}\frac{1}{2}\|w_{t,\widehat{\eta}}-w_{\mathrm{train}}\|^2_{H_{\mathrm{train}}}\mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widehat{\eta})\right\} \leqslant \frac{\epsilon^2 d\sigma^2}{n}$. The other case can be resolved using techniques in the proof of Lemma D.2.9.

Recall the population meta objective

$$F_{TbT}(\eta) = \mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}} + \frac{n-d}{2n}\sigma^2.$$

Therefore, we only need to construct an ϵ-net for the first term.

We can divide $\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}}$ as follows,

$$\begin{aligned}
&\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}} \\
=&\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_2(\widehat{\eta})\right\} + \mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widehat{\eta})\right\} \\
&+ \mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\bar{\mathcal{E}}_1\right\}.
\end{aligned}$$

We will construct an ϵ-net for the first term and show the other two terms are small. Let's first consider the third term. Assuming $n \geqslant 40d$, we know $\Pr[\mathcal{E}_1] \leqslant \exp(-\Omega(n))$. Since $\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}}$ is $O(1)$-subexponential, by Cauchy-Schwarz inequality, we have $\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\bar{\mathcal{E}}_1\right\} = O(1)\exp(-\Omega(n))$. Choosing $n \geqslant c\log(n/(\epsilon d))$ for some constant c, we know $\frac{1}{2}\|w_{t,\widehat{\eta}} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\bar{\mathcal{E}}_1\right\} \leqslant \frac{\epsilon^2 d\sigma^2}{n}$.

Then we upper bound the second term. Since $\mathbb{E}\frac{1}{2}\|w_{t,\widehat{\eta}} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widehat{\eta})\right\} \leqslant \frac{\epsilon^2 d\sigma^2}{n}$ and $\frac{1}{2}\|w_{t,\widehat{\eta}} - w_{\text{train}}\|^2_{H_{\text{train}}} \geqslant \frac{35^2\sigma^2}{4}$ when $w_{t,\widehat{\eta}}$ diverges, we know $\Pr\left[\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widehat{\eta})\right] \leqslant \frac{4\epsilon^2 d}{35^2 n}$. Then, we can upper bound the second term as follows,

$$\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1 \cap \bar{\mathcal{E}}_2(\widehat{\eta})\right\} \leqslant \frac{3 \times 45^2\sigma^2}{4}\frac{4\epsilon^2 d}{35^2 n} \leqslant \frac{6\epsilon^2 d\sigma^2}{n}$$

Next, similar as in Lemma D.2.9, we can show that the first term, which is $\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_2(\widehat{\eta})\right\}$, is $O(t)$-lipschitz. Therefore, there exists an $\frac{\epsilon^2 d\sigma^2}{n}$-net N for $\mathbb{E}\frac{1}{2}\|w_{t,\eta} - w_{\text{train}}\|^2_{H_{\text{train}}} \mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_2(\widehat{\eta})\right\}$ with size $O(\frac{tn}{\epsilon^2 d})$. That means, for any

$\eta \in [0, \widehat{\eta}]$,

$$\left| \mathbb{E} \frac{1}{2} \|w_{t,\eta} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_2(\widehat{\eta})\right\} - \mathbb{E} \frac{1}{2} \|w_{t,\eta'} - w_{\text{train}}\|_{H_{\text{train}}}^2 \mathbb{1}\left\{\mathcal{E}_1 \cap \mathcal{E}_2(\widehat{\eta})\right\} \right| \leqslant \frac{\epsilon^2 d\sigma^2}{n}$$

for $\eta' = \arg\min_{\eta'' \in N, \eta'' \leqslant \eta} (\eta - \eta'')$.

Combing with the upper bounds on the second term and the third term, we have for any $\eta \in [0, \widehat{\eta}]$,

$$|F_{TbT}(\eta) - F_{TbT}(\eta')| \leqslant \frac{8\epsilon^2 d\sigma^2}{n}$$

for $\eta' = \arg\min_{\eta'' \in N, \eta'' \leqslant \eta} (\eta - \eta'')$. □

Proof of Lemma D.3.10. By Lemma D.3.7, we know $\widehat{\eta}$ is a constant. For each $k \in [m]$, let $\mathcal{E}_{1,k}$ be the event that $\sqrt{n}/2 \leqslant \sigma_i(X_{\text{train}}^{(k)}) \leqslant 3\sqrt{n}/2$ and $1/2 \leqslant \lambda_i(H_{\text{train}}^{(k)}) \leqslant 3/2$ for all $i \in [d]$ and $\sqrt{n}\sigma/2 \leqslant \left\|\xi_{\text{train}}^{(k)}\right\| \leqslant 2\sqrt{n}\sigma$. Assuming $n \geqslant 40d$, by Lemma D.3.4, we know with probability at least $1 - m\exp(-\Omega(n))$, $\mathcal{E}_{1,k}$'s hold for all $k \in [m]$.

Then, similar as in Lemma D.2.10, there exists an $\frac{\epsilon^2 d\sigma^2}{n}$-net N' with $|N'| = O(\frac{nt}{\epsilon^2 d} + m)$ for $\widehat{F}_{TbT}$. That means, for any $\eta \in [0, \widehat{\eta}]$,

$$\left| \widehat{F}_{TbT}(\eta) - \widehat{F}_{TbT}(\eta') \right| \leqslant \frac{\epsilon^2 d\sigma^2}{n}$$

for $\eta' = \arg\min_{\eta'' \in N', \eta'' \leqslant \eta} (\eta - \eta'')$. □

D.4 Proofs of Train-by-Train v.s. Train-by-Validation (SGD)

Previously, we have shown that train-by-validation generalizes better than train-by-train when the tasks are trained by GD and when the number of samples is small. In this section, we show a similar phenomenon also appears in the SGD setting.

In the train-by-train setting, each task P contains a training set $S_{\text{train}} = \{(x_i, y_i)\}_{i=1}^n$.

The inner objective is defined as $\widehat{f}(w) = \frac{1}{2n}\sum_{(x,y)\in S_{\text{train}}}\left(\langle w\rangle x - y\right)^2$. Let $\{w_{\tau,\eta}\}$ be the SGD sequence running on $\widehat{f}(w)$ from initialization 0 (without truncation). Thus, $w_{\tau,\eta} = w_{\tau-1,\eta} - \eta\widehat{\nabla}\widehat{f}(w_{\tau-1,\eta})$, where $\widehat{\nabla}\widehat{f}(w_{\tau-1,\eta}) = \left(\langle w_{\tau-1,\eta}\rangle x_{i(\tau-1)} - y_{i(\tau-1)}\right)x_{i(\tau-1)}$. Here index $i(\tau-1)$ is independently and uniformly sampled from $[n]$. We denote the SGD noise as $n_{\tau-1,\eta} := \widehat{\nabla}\widehat{f}(w_{\tau-1,\eta}) - \nabla\widehat{f}(w_{\tau-1,\eta})$. The meta-loss on task P is defined as follows,

$$\Delta_{TbT(n)}(\eta, P) = \mathbb{E}_{\text{SGD}}\widehat{f}(w_{t,\eta}) = \mathbb{E}_{\text{SGD}}\frac{1}{2n}\sum_{(x,y)\in S_{\text{train}}}\left(\langle w_{t,\eta}\rangle x - y\right)^2,$$

where the expectation is taken over the SGD noise. Note $w_{t,\eta}$ depends on the SGD noise along the trajectory. Then, the empirical meta objective $\widehat{F}_{TbT(n)}(\eta)$ is the average of the meta-loss across m different specific tasks

$$\widehat{F}_{TbT(n)}(\eta) = \frac{1}{m}\sum_{k=1}^{m}\Delta_{TbT(n)}(\eta, P_k). \tag{D.2}$$

In order to control the SGD noise in expectation, we restrict the feasible set of step sizes into $O(1/d)$. We show within this range, the optimal step size under $\widehat{F}_{TbT(n)}$ is $\Omega(1/d)$ and the learned weight is far from ground truth w^* on new tasks. We prove Theorem D.4.1 in Section D.4.1.

Theorem D.4.1. *Let the meta objective $\widehat{F}_{TbT(n)}$ be as defined in Equation D.2 with $n \in [d/4, 3d/4]$. Suppose σ is a constant. Assume unroll length $t \geqslant c_2 d$ and dimension $d \geqslant c_4\log(m)$ for certain constants c_2, c_4. Then, with probability at least 0.99 in the sampling of training tasks $P_1, \cdots, P_m$ and test task P,*

$$\eta^*_{train} = \Omega(1/d) \text{ and } \mathbb{E}_{SGD}\left\|w_{t,\eta^*_{train}} - w^*\right\|^2 = \Omega(\sigma^2),$$

for all $\eta^*_{train} \in \arg\min_{0 \leqslant \eta \leqslant \frac{1}{2L^3 d}} \widehat{F}_{TbT(n)}(\eta)$, *where* $L = 100$ *and* $w_{t,\eta^*_{train}}$ *is trained by running SGD on test task* P.

In the train-by-validation setting, each task P contains a training set S_{train} with n_1 samples and a validation set with n_2 samples. The inner objective is defined as $\widehat{f}(w) = \frac{1}{2n_1} \sum_{(x,y) \in S_{\text{train}}} (\langle w \rangle x - y)^2$. Let $\{w_{\tau,\eta}\}$ be the SGD sequence running on $\widehat{f}(w)$ from initialization 0 (with the same truncation defined in Section 5.4). For each task P, the meta-loss $\Delta_{TbV(n_1,n_2)}(\eta, P)$ is defined as

$$\Delta_{TbV(n_1,n_2)}(\eta, P) = \mathbb{E}_{\text{SGD}} \frac{1}{2n_2} \sum_{(x,y) \in S_{\text{valid}}} (\langle w_{t,\eta} \rangle x - y)^2 .$$

The empirical meta objective $\widehat{F}_{TbV(n_1,n_2)}(\eta)$ is the average of the meta-loss across m different tasks $P_1, P_2, ..., P_m$,

$$\widehat{F}_{TbV(n_1,n_2)}(\eta) = \frac{1}{m} \sum_{k=1}^{m} \Delta_{TbV(n_1,n_2)}(\eta, P_k). \tag{D.3}$$

In order to bound the SGD noise with high probability, we restrict the feasible set of the step sizes into $O(\frac{1}{d^2 \log^2 d})$. Within this range, we prove the optimal step size under $\widehat{F}_{TbV(n_1,n_2)}$ is $\Theta(1/t)$ and the learned weight is better than initialization 0 by a constant on new tasks. Theorem D.4.2 is proved in Section D.4.2.

Theorem D.4.2. *Let the meta objective* $\widehat{F}_{TbV(n_1,n_2)}$ *be as defined in Equation D.3 with* $n_1, n_2 \in [d/4, 3d/4]$. *Assume noise level* σ *is a large constant* c_1. *Assume unroll length* $t \geqslant c_2 d^2 \log^2(d)$, *number of training tasks* $m \geqslant c_3$ *and dimension* $d \geqslant c_4$ *for certain constants* c_2, c_3, c_4. *There exists constant* c_5 *such that with probability at least*

0.99 *in the sampling of training tasks, we have*

$$\eta^*_{valid} = \Theta(1/t) \text{ and } \mathbb{E}\left\|w_{t,\eta^*_{valid}} - w^*\right\|^2 = \|w^*\|^2 - \Omega(1)$$

for all $\eta^*_{valid} \in \arg\min_{0\leqslant\eta\leqslant\frac{1}{c_5 d^2\log^2(d)}} \widehat{F}_{TbV(n_1,n_2)}(\eta)$*, where the expectation is taken over the new tasks and SGD noise.*

Notations: In the following proofs, we use the same set of notations defined in Appendix D.2. We use $\mathbb{E}_{P\sim\mathcal{T}}$ to denote the expectation over the sampling of tasks and use $\mathbb{E}_{\text{SGD}}$ to denote the expectation over the SGD noise. We use $\mathbb{E}$ to denote $\mathbb{E}_{P\sim\mathcal{T}}\mathbb{E}_{\text{SGD}}$. Same as in Appendix D.2, we use letter L to denote constant 100, which upper bounds $\|H_{\text{train}}\|$ with high probability.

D.4.1 Train-by-Train (SGD)

Recall Theorem D.4.1 as follows.

Theorem D.4.1. *Let the meta objective* $\widehat{F}_{TbT(n)}$ *be as defined in Equation D.2 with* $n \in [d/4, 3d/4]$*. Suppose* σ *is a constant. Assume unroll length* $t \geqslant c_2 d$ *and dimension* $d \geqslant c_4 \log(m)$ *for certain constants* c_2, c_4*. Then, with probability at least* 0.99 *in the sampling of training tasks* $P_1, \cdots, P_m$ *and test task* P*,*

$$\eta^*_{train} = \Omega(1/d) \text{ and } \mathbb{E}_{SGD}\left\|w_{t,\eta^*_{train}} - w^*\right\|^2 = \Omega(\sigma^2),$$

for all $\eta^*_{train} \in \arg\min_{0\leqslant\eta\leqslant\frac{1}{2L^3 d}} \widehat{F}_{TbT(n)}(\eta)$*, where* $L = 100$ *and* $w_{t,\eta^*_{train}}$ *is trained by running SGD on test task* P*.*

In order to prove Theorem D.4.1, we first show that η^*_{train} is $\Omega(1/d)$ in Lemma D.4.3. The proof is similar as in the GD setting. As long as $\eta = O(1/d)$, the SGD noise

is dominated by the full gradient. Then, we can show that $\Delta_{TbT}(\eta, P)$ is roughly $(1-\Theta(1)\eta)^t$, which implies that $\eta^*_{\text{train}} = \Omega(1/d)$. We leave the proof of Lemma D.4.3 into Section D.4.1.

Lemma D.4.3. *Assume $t \geqslant c_2 d$ with certain constant c_2. With probability at least $1 - m\exp(-\Omega(d))$ in the sampling of m training tasks,*

$$\eta^*_{train} \geqslant \frac{1}{6L^5 d},$$

for all $\eta^*_{train} \in \arg\min_{0\leqslant\eta\leqslant\frac{1}{2L^3 d}} \widehat{F}_{TbT}(\eta)$.

Let $P = (\mathcal{D}(w^*), S_{\text{train}}, \ell)$ be an independently sampled test task with $|S_{\text{train}}| = n \in [d/4, 3d/4]$. For any step size $\eta \in [\frac{1}{6L^5 d}, \frac{1}{2L^3 d}]$, let $w_{t,\eta}$ be the weight obtained by running SGD on $\widehat{f}(w)$ for t steps. Next, we show $\mathbb{E}_{\text{SGD}}\|w_{t,\eta} - w^*\|^2 = \Omega(\sigma^2)$ with high probability in the sampling of P.

Lemma D.4.4. *Suppose σ is a constant. Assume unroll length $t \geqslant c_2 d$ for some constant c_2. With probability at least $1 - \exp(-\Omega(d))$ in the sampling of test task P,*

$$\mathbb{E}_{SGD}\|w_{t,\eta} - w^*\|^2 \geqslant \frac{\sigma^2}{128L},$$

for all $\eta \in [\frac{1}{6L^5 d}, \frac{1}{2L^3 d}]$, where $w_{t,\eta}$ is obtained by running SGD on task P for t iterations.

With Lemma Lemma D.4.3 and Lemma D.4.4, the proof of Theorem D.4.1 is straightforward.

Proof of Theorem D.4.1. Combing Lemma D.4.3 and Lemma D.4.4, we know as long as σ is a constant, $t \geqslant c_2 d, d \geqslant c_4 \log(m)$, with probability at least 0.99, $\eta^*_{\text{train}} = \Omega(1/d)$ and $\mathbb{E}_{\text{SGD}}\left\|w_{t,\eta^*_{\text{train}}} - w^*\right\|^2 = \Omega(\sigma^2)$, for all $\eta^*_{\text{train}} \in \arg\min_{0\leqslant\eta\leqslant\frac{1}{2L^3 d}} \widehat{F}_{TbT}(\eta)$. □

Proof of Lemma D.4.3. The proof is very similar to the proof of Lemma 5.4.3 except that we need to bound the SGD noise term. For each $k \in [m]$, let $\mathcal{E}_k$ be the event that $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\text{train}}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{\text{train}}\| \leqslant \sqrt{d}\sigma$. According to Lemma D.2.3 and Lemma D.5.1, we know for each $k \in [m]$, $\mathcal{E}_k$ happens with probability at least $1 - \exp(-\Omega(d))$. Taking a union bound over all $k \in [m]$, we know $\cap_{k\in[m]}\mathcal{E}_k$ holds with probability at least $1 - m\exp(-\Omega(d))$. From now on, we assume $\cap_{k\in[m]}\mathcal{E}_k$ holds.

For each $k \in [m]$, we have

$$\Delta_{TbT}(\eta, P_k) := \frac{1}{2}\mathbb{E}_{\text{SGD}}\left\|w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|_{H_{\text{train}}^{(k)}}^2.$$

Since $1/L \leqslant \lambda_i(H_{\text{train}}^{(k)}) \leqslant L$ and $(w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)})$ is in the span of $H_{\text{train}}^{(k)}$, we have

$$\frac{1}{2L}\mathbb{E}_{\text{SGD}}\left\|w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2 \leqslant \Delta_{TbT}(\eta, P_k) \leqslant \frac{L}{2}\mathbb{E}_{\text{SGD}}\left\|w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2.$$

Recall the updates of stochastic gradient descent,

$$w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)} = (I - \eta H_{\text{train}}^{(k)})(w_{t-1,\eta}^{(k)} - w_{\text{train}}^{(k)}) - \eta n_{t-1,\eta}^{(k)}.$$

Therefore,

$$\begin{aligned}
&\mathbb{E}_{\text{SGD}}\left[\left\|w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2 | w_{t-1,\eta}^{(k)}\right] \\
=&\left\|(I - \eta H_{\text{train}}^{(k)})(w_{t-1,\eta}^{(k)} - w_{\text{train}}^{(k)})\right\|^2 + \eta^2\mathbb{E}_{\text{SGD}}\left[\left\|n_{t-1,\eta}^{(k)}\right\|^2 | w_{t-1,\eta}^{(k)}\right].
\end{aligned}$$

We know for any $\eta \leqslant 1/L$,

$$\begin{aligned}
&(1-2\eta L)\left\|w_{t-1,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2 \\
\leqslant&\left\|(I-\eta H_{\text{train}}^{(k)})(w_{t-1,\eta}^{(k)} - w_{\text{train}}^{(k)})\right\|^2 \leqslant (1-\frac{\eta}{L})\left\|w_{t-1,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2.
\end{aligned}$$

The noise can be bounded as follows,

$$\begin{aligned}
&\eta^2\mathbb{E}_{\text{SGD}}\left[\left\|n_{t-1,\eta}^{(k)}\right\|^2 | w_{t-1,\eta}^{(k)}\right] \\
=&\eta^2\mathbb{E}_{\text{SGD}}\left[\left\|x_{i(t-1)}x_{i(t-1)}^\top(w_{t-1,\eta}^{(k)} - w_{\text{train}}^{(k)}) - H_{\text{train}}^{(k)}(w_{t-1,\eta}^{(k)} - w_{\text{train}}^{(k)})\right\|^2 | w_{t-1,\eta}^{(k)}\right] \\
\leqslant&\eta^2\mathbb{E}_{\text{SGD}}\left[\left\|x_{i(t-1)}x_{i(t-1)}^\top(w_{t-1,\eta}^{(k)} - w_{\text{train}}^{(k)})\right\|^2 | w_{t-1,\eta}^{(k)}\right] \\
\leqslant&\eta^2 \max_{i(t-1)}\left\|x_{i(t-1)}\right\|^2\left\|w_{t-1,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|_{H_{\text{train}}^{(k)}}^2.
\end{aligned}$$

Since $\|X_{\text{train}}\| \leqslant \sqrt{L}\sqrt{d}$, we immediately know $\max_{i(t-1)}\left\|x_{i(t-1)}\right\| \leqslant \sqrt{L}\sqrt{d}$. Therefore, we can bound the noise as follows,

$$\begin{aligned}
\eta^2\mathbb{E}_{\text{SGD}}\left[\left\|n_{t-1,\eta}^{(k)}\right\|^2 | w_{t-1,\eta}^{(k)}\right] \leqslant&\eta^2 \max_{i(t-1)}\left\|x_{i(t-1)}\right\|^2\left\|w_{t-1,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|_{H_{\text{train}}^{(k)}}^2 \\
\leqslant&L^2\eta^2 d\left\|w_{t-1,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2.
\end{aligned}$$

As long as $\eta \leqslant \frac{1}{2L^3 d}$, we have

$$(1-\eta L)\left\|w_{t-1,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2 \leqslant \mathbb{E}_{\text{SGD}}\left[\left\|w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2 | w_{t-1,\eta}^{(k)}\right] \leqslant (1-\frac{\eta}{2L})\left\|w_{t-1,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2.$$

This further implies

$$(1-\eta L)^t\|w_{\text{train}}\|^2 \leqslant \mathbb{E}_{\text{SGD}}\left\|w_{t,\eta}^{(k)} - w_{\text{train}}^{(k)}\right\|^2 \leqslant (1-\frac{\eta}{2L})^t\|w_{\text{train}}\|^2.$$

Let $\eta_2 := \frac{1}{2L^3 d}$, we have

$$\Delta_{TbT}(\eta, P_k) \leqslant \frac{L}{2}(1 - \frac{1}{4L^4 d})^t \|w_{\text{train}}\|^2$$

Let $\eta_1 := \frac{1}{6L^5 d}$, for all $\eta \in [0, \eta_1]$ we have

$$\Delta_{TbT}(\eta, P_k) \geqslant \frac{1}{2L}(1 - \frac{1}{6L^4 d})^t \|w_{\text{train}}\|^2.$$

As long as $t \geqslant c_2 d$ for certain constant c_2, we know

$$\frac{1}{2L}(1 - \frac{1}{6L^4 d})^t \|w_{\text{train}}\|^2 > \frac{L}{2}(1 - \frac{1}{4L^4 d})^t \|w_{\text{train}}\|^2.$$

As this holds for all $k \in [m]$ and $\widehat{F}_{TbT} = 1/m \sum_{i=1}^m \Delta_{TbT}(\eta, P_k)$, we know the optimal step size η^*_{train} is within $[\frac{1}{6L^5 d}, \frac{1}{2L^3 d}]$. □

We rely the following technical lemma to prove Lemma D.4.4.

Lemma D.4.5. *Suppose σ is a constant. Given any $\epsilon > 0$, with probability at least $1 - O(1/\epsilon)\exp(-\Omega(\epsilon^2 d))$,*

$$\left|\left\langle B_{t,\eta} w^*_{train} - w^* \right\rangle B_{t,\eta} (X_{train})^\dagger \xi_{train}\right| \leqslant \epsilon,$$

for all $\eta \in [0, \frac{1}{2L^3 d}]$.

Proof of Lemma D.4.5. By Lemma D.2.3, with probability at least $1 - \exp(-\Omega(d))$, $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\text{train}}) \leqslant L$ for all $i \in [n]$. Therefore $\left\|[(X_{\text{train}})^\dagger]^\top B_{t,\eta}(B_{t,\eta} w^*_{\text{train}} - w^*)\right\| \leqslant 2\sqrt{L}/\sqrt{d}$. Notice that ξ_{train} is independent with $[(X_{\text{train}})^\dagger]^\top B_{t,\eta}(B_{t,\eta} w^*_{\text{train}} - w^*)$. By Hoeffding's inequality, with probability at least $1 - \exp(-\Omega(\epsilon^2 d))$,

$$\left|\left\langle [(X_{\text{train}})^\dagger]^\top B_{t,\eta}(B_{t,\eta} w^*_{\text{train}} - w^*) \right\rangle \xi_{\text{train}}\right| \leqslant \epsilon.$$

Next, we construct an ϵ-net for η and show the crossing term is small for all $\eta \in [0, \frac{1}{2L^3 d}]$. For simplicity, denote $g(\eta) := \langle B_{t,\eta} w^*_{\text{train}} - w^* \rangle B_{t,\eta} (X_{\text{train}})^\dagger \xi_{\text{train}}$. Taking the derivative of $g(\eta)$, we have

$$\begin{aligned} g'(\eta) =& t \left\langle H_{\text{train}} (I - \eta H_{\text{train}})^{t-1} w^*_{\text{train}} \right\rangle B_{t,\eta} (X_{\text{train}})^\dagger \xi_{\text{train}} \\ & + t \langle B_{t,\eta} w^*_{\text{train}} - w^* \rangle H_{\text{train}} (I - \eta H_{\text{train}})^{t-1} (X_{\text{train}})^\dagger \xi_{\text{train}} \end{aligned}$$

According to Lemma D.5.1, we know with probability at least $1 - \exp(-\Omega(d))$, $\|\xi_{\text{train}}\| \leqslant \sqrt{d}\sigma$. Therefore, the derivative $g'(\eta)$ can be bounded as follows,

$$|g'(\eta)| = O(1) t (1 - \frac{\eta}{L})^{t-1}$$

Similar as in Lemma D.2.15, there exists an ϵ-net N_ϵ with size $O(1/\epsilon)$ such that for any $\eta \in [0, \frac{1}{3L^3 d}]$, there exists $\eta' \in N_\epsilon$ with $|g(\eta) - g(\eta')| \leqslant \epsilon$. Taking a union bound over N_ϵ, we have with probability at least $1 - O(1/\epsilon) \exp(-\Omega(\epsilon^2 d))$, for every $\eta \in N_\epsilon$,

$$\left| \langle B_{t,\eta} w^*_{\text{train}} - w^* \rangle B_{t,\eta} (X_{\text{train}})^\dagger \xi_{\text{train}} \right| \leqslant \epsilon.$$

which implies for every $\eta \in [0, \frac{1}{3L^3 d}]$.

$$\left| \langle B_{t,\eta} w^*_{\text{train}} - w^* \rangle B_{t,\eta} (X_{\text{train}})^\dagger \xi_{\text{train}} \right| \leqslant 2\epsilon.$$

Changing ϵ to $\epsilon'/2$ finishes the proof. □

Proof of Lemma D.4.4. According to Lemma D.2.3 and Lemma D.5.1, we know with probability at least $1 - \exp(-\Omega(d))$, $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\text{train}}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{\text{train}}\| \leqslant \sqrt{d}\sigma$. We assume these properties hold in the proof and take a union bound at the end.

Recall that $\mathbb{E}_{\mathrm{SGD}}\|w_{t,\eta} - w^*\|^2$ can be lower bounded as follows,

$$\begin{aligned}
&\mathbb{E}_{\mathrm{SGD}}\|w_{t,\eta} - w^*\|^2 \\
=&\mathbb{E}_{\mathrm{SGD}}\left\|B_{t,\eta}(w^*_{\mathrm{train}} + (X_{\mathrm{train}})^\dagger \xi_{\mathrm{train}}) - \eta\sum_{\tau=0}^{t-1}(I - \eta H_{\mathrm{train}})^{t-1-\tau} n_{\tau,\eta} - w^*\right\|^2 \\
\geqslant&\left\|B_{t,\eta}(w^*_{\mathrm{train}} + (X_{\mathrm{train}})^\dagger \xi_{\mathrm{train}}) - w^*\right\|^2 \\
\geqslant&\left\|B_{t,\eta}(X_{\mathrm{train}})^\dagger \xi_{\mathrm{train}}\right\|^2 + 2\left\langle B_{t,\eta} w^*_{\mathrm{train}} - w^*\right\rangle B_{t,\eta}(X_{\mathrm{train}})^\dagger \xi_{\mathrm{train}}
\end{aligned}$$

For any $\eta \in [\frac{1}{6L^5 d}, \frac{1}{2L^3 d}]$, we can lower bound the first term as follows,

$$\begin{aligned}
\left\|B_{t,\eta}(X_{\mathrm{train}})^\dagger \xi_{\mathrm{train}}\right\|^2 \geqslant& \left(1 - \exp\left(-\frac{\eta t}{L}\right)\right)^2 \frac{\sigma^2}{16L} \\
\geqslant& \left(1 - \exp\left(-\frac{t}{6L^6 d}\right)\right)^2 \frac{\sigma^2}{16L} \\
\geqslant& \frac{\sigma^2}{64L},
\end{aligned}$$

where the last inequality holds as long as $t \geqslant c_2 d$ for certain constant c_2.

Choosing $\epsilon = \frac{\sigma^2}{256L}$ in Lemma D.4.5, we know with probability at least $1 - \exp(-\Omega(d))$,

$$\left|\left\langle B_{t,\eta} w^*_{\mathrm{train}} - w^*\right\rangle B_{t,\eta}(X_{\mathrm{train}})^\dagger \xi_{\mathrm{train}}\right| \leqslant \frac{\sigma^2}{256L},$$

for all $\eta \in [0, \frac{1}{2L^3 d}]$.

Overall, we have $\mathbb{E}_{\mathrm{SGD}}\|w_{t,\eta} - w^*\|^2 \geqslant \frac{\sigma^2}{128L}$. Taking a union bound over all the bad events, we know this happens with probability at least $1 - \exp(-\Omega(d))$. □

D.4.2 Train-by-Validation (SGD)

Recall Theorem D.4.2 as follows.

Theorem D.4.2. *Let the meta objective $\widehat{F}_{TbV(n_1,n_2)}$ be as defined in Equation D.3*

with $n_1, n_2 \in [d/4, 3d/4]$. *Assume noise level* σ *is a large constant* c_1. *Assume unroll length* $t \geqslant c_2 d^2 \log^2(d)$, *number of training tasks* $m \geqslant c_3$ *and dimension* $d \geqslant c_4$ *for certain constants* c_2, c_3, c_4. *There exists constant* c_5 *such that with probability at least* 0.99 *in the sampling of training tasks, we have*

$$\eta^*_{valid} = \Theta(1/t) \text{ and } \mathbb{E}\left\| w_{t,\eta^*_{valid}} - w^* \right\|^2 = \|w^*\|^2 - \Omega(1)$$

for all $\eta^*_{valid} \in \arg\min_{0 \leqslant \eta \leqslant \frac{1}{c_5 d^2 \log^2(d)}} \widehat{F}_{TbV(n_1,n_2)}(\eta)$, *where the expectation is taken over the new tasks and SGD noise.*

To prove Theorem D.4.2, we first study the behavior of the population meta objective F_{TbV}. That is,

$$\begin{aligned} F_{TbV}(\eta) := \mathbb{E}_{P\sim\mathcal{T}} \Delta_{TbV}(\eta, P) =& \mathbb{E}_{P\sim\mathcal{T}} \mathbb{E}_{\mathrm{SGD}} \frac{1}{2} \left\| w_{t,\eta} - w^* - (X_{\mathrm{valid}})^\dagger \xi_{\mathrm{valid}} \right\|^2_{H_{\mathrm{valid}}} \\ =& \mathbb{E}_{P\sim\mathcal{T}} \mathbb{E}_{\mathrm{SGD}} \frac{1}{2} \|w_{t,\eta} - w^*\|^2 + \frac{\sigma^2}{2}. \end{aligned}$$

We show that the optimal step size for the population meta objective F_{TbV} is $\Theta(1/t)$ and $\mathbb{E}_{P\sim\mathcal{T}} \mathbb{E}_{\mathrm{SGD}} \|w_{t,\eta} - w^*\|^2 = \|w^*\|^2 - \Omega(1)$ under the optimal step size.

Lemma D.4.6. *Suppose* σ *is a large constant* c_1. *Assume* $t \geqslant c_2 d^2 \log^2(d), d \geqslant c_4$ *for some constants* c_2, c_4. *There exist* $\eta_1, \eta_2, \eta_3 = \Theta(1/t)$ *with* $\eta_1 < \eta_2 < \eta_3$ *and constant* c_5 *such that*

$$\begin{aligned} F_{TbV}(\eta_2) &\leqslant \frac{1}{2}\|w^*\|^2 - \frac{9}{10}C + \frac{\sigma^2}{2} \\ F_{TbV}(\eta) &\geqslant \frac{1}{2}\|w^*\|^2 - \frac{6}{10}C + \frac{\sigma^2}{2}, \forall \eta \in [0, \eta_1] \cup [\eta_3, \frac{1}{c_5 d^2 \log^2(d)}] \end{aligned}$$

where C *is a positive constant.*

In order to relate the behavior of F_{TbV} to $\widehat{F}_{TbV}$, we show a generalization result

from $\widehat{F}_{TbV}$ to F_{TbV} for $\eta \in [0, \frac{1}{c_5 d^2 \log^2(d/\epsilon)}]$.

Lemma D.4.7. *For any $1 > \epsilon > 0$, assume σ is a constant and $d \geqslant c_4 \log(1/\epsilon)$ for some constant c_4. There exists constant c_5 such that with probability at least $1 - O(1/\epsilon)\exp(-\Omega(\epsilon^2 m))$,*

$$|\widehat{F}_{TbV}(\eta) - F_{TbV}(\eta)| \leqslant \epsilon,$$

for all $\eta \in [0, \frac{1}{c_5 d^2 \log^2(d/\epsilon)}]$.

Combining Lemma D.4.6 and Lemma D.4.7, we give the proof of Theorem D.4.2.

Proof of Theorem D.4.2. The proof is almost the same as in the GD setting (Theorem D.2.2). We omit the details here. □

Behavior of F_{TbV} for $\eta \in [0, \frac{1}{c_5 d^2 \log^2 d}]$

In this section, we give the proof of Lemma D.4.6. Recall the lemma as follows,

Lemma D.4.6. *Suppose σ is a large constant c_1. Assume $t \geqslant c_2 d^2 \log^2(d), d \geqslant c_4$ for some constants c_2, c_4. There exist $\eta_1, \eta_2, \eta_3 = \Theta(1/t)$ with $\eta_1 < \eta_2 < \eta_3$ and constant c_5 such that*

$$F_{TbV}(\eta_2) \leqslant \frac{1}{2}\|w^*\|^2 - \frac{9}{10}C + \frac{\sigma^2}{2}$$
$$F_{TbV}(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{6}{10}C + \frac{\sigma^2}{2}, \forall \eta \in [0, \eta_1] \cup [\eta_3, \frac{1}{c_5 d^2 \log^2(d)}]$$

where C is a positive constant.

Recall that $F_{TbV}(\eta) = \mathbb{E}_{P\sim\mathcal{T}}\mathbb{E}_{\mathrm{SGD}} 1/2\|w_{t,\eta} - w^*\|^2 + \sigma^2/2$. We denote $Q(\eta) := \mathbb{E}_{\mathrm{SGD}} 1/2\|w_{t,\eta} - w^*\|^2$. Recall that we truncate the SGD sequence once the weight norm exceeds $4\sqrt{L}\sigma$. Due to the truncation, the expectation of $1/2\|w_{t,\eta} - w^*\|^2$ over SGD noise is very tricky to analyze.

Instead, we define an auxiliary sequence $\{w'_{\tau,\eta}\}$ that is obtained by running SGD on task P without truncation and we first study $Q'(\eta) := 1/2\mathbb{E}_{\text{SGD}}\left\|w'_{t,\eta} - w^*\right\|^2$. In Lemma D.4.8, we show that with high probability in the sampling of task P, the minimizer of $Q'(\eta)$ is $\Theta(1/t)$. The proof is very similar as the proof of Lemma D.2.14 except that we need to bound the SGD noise at step size η_2. We defer the proof into Section D.4.2.

Lemma D.4.8. *Given a task P, let $\{w'_{\tau,\eta}\}$ be the weight obtained by running SGD on task P without truncation. Choose σ as a large constant c_1. Assume unroll length $t \geqslant c_2 d$ for some constant c_2. With probability at least $1 - \exp(-\Omega(d))$ over the sampling of task P, $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{train}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{train}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{train}\| \leqslant \sqrt{d}\sigma$ and there exists $\eta_1, \eta_2, \eta_3 = \Theta(1/t)$ with $\eta_1 < \eta_2 < \eta_3$ such that*

$$Q'(\eta_2) := 1/2\mathbb{E}_{SGD}\left\|w'_{t,\eta_2} - w^*\right\|^2 \leqslant \frac{1}{2}\|w^*\|^2 - C$$

$$Q'(\eta) := 1/2\mathbb{E}_{SGD}\left\|w'_{t,\eta} - w^*\right\|^2 \geqslant \frac{1}{2}\|w^*\|^2 - \frac{C}{2}, \forall \eta \in [0, \eta_1] \cup [\eta_3, 1/L]$$

where C is a positive constant.

To relate the behavior of $Q'(\eta)$ defined on $\{w'_{\tau,\eta}\}$ to the behavior of $Q(\eta)$ defined on $\{w_{\tau,\eta}\}$. We show when the step size is small enough, the SGD sequence gets truncated with very small probability so that sequence $\{w_{\tau,\eta}\}$ almost always coincides with sequence $\{w'_{\tau,\eta}\}$. The proof of Lemma D.4.9 is deferred into Section D.4.2.

Lemma D.4.9. *Given a task P, assume $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{train}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{train}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{train}\| \leqslant \sqrt{d}\sigma$. Given any $\epsilon > 0$, suppose*

$\eta \leqslant \frac{1}{c_5 d^2 \log^2(d/\epsilon)}$ *for some constant* c_5*, we have*

$$|Q(\eta) - Q'(\eta)| \leqslant \epsilon.$$

Combining Lemma D.4.8 and Lemma D.4.9, we give the proof of lemma D.4.6.

Proof of Lemma D.4.6. Recall that we define $Q(\eta) := 1/2\mathbb{E}_{\mathrm{SGD}}\|w_{t,\eta} - w^*\|^2$ and $Q'(\eta) = 1/2\mathbb{E}_{\mathrm{SGD}}\|w'_{t,\eta} - w^*\|^2$. Here, $\{w'_{\tau,\eta}\}$ is a SGD sequence running on task P without truncation.

According to Lemma D.4.8, with probability at least $1 - \exp(-\Omega(d))$ over the sampling of task P, $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\mathrm{train}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\mathrm{train}}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{\mathrm{train}}\| \leqslant \sqrt{d}\sigma$ and there exists $\eta_1, \eta_2, \eta_3 = \Theta(1/t)$ with $\eta_1 < \eta_2 < \eta_3$ such that

$$Q'(\eta_2) \leqslant \frac{1}{2}\|w^*\|^2 - C$$

$$Q'(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{C}{2}, \forall \eta \in [0, \eta_1] \cup [\eta_3, 1/L]$$

where C is a positive constant. Call this event $\mathcal{E}$. Suppose the probability that $\mathcal{E}$ happens is $1 - \delta$. We can write $\mathbb{E}_{P\sim\mathcal{T}}Q(\eta)$ as follows,

$$\mathbb{E}_{P\sim\mathcal{T}}Q(\eta) = \mathbb{E}_{P\sim\mathcal{T}}[Q(\eta)|\mathcal{E}]\Pr[\mathcal{E}] + \mathbb{E}_{P\sim\mathcal{T}}[Q(\eta)|\bar{\mathcal{E}}]\Pr[\bar{\mathcal{E}}].$$

According to the algorithm, we know $\|w_{t,\eta}\|$ is always bounded by $4\sqrt{L}\sigma$. Therefore, $Q(\eta) := 1/2\|w_{t,\eta} - w^*\|^2 \leqslant 13L\sigma^2$. By Lemma D.4.9, we know conditioning on $\mathcal{E}$, $|Q(\eta) - Q'(\eta)| \leqslant \epsilon$ for any $\eta \leqslant \frac{1}{c_5 d^2 \log^2(d/\epsilon)}$. As long as $t \geqslant c_2 d^2 \log^2(d/\epsilon)$ for certain constant c_2, we know $\eta_3 \leqslant \frac{1}{c_5 d^2 \log^2(d/\epsilon)}$.

When $\eta = \eta_2$, we have

$$\begin{aligned}\mathbb{E}_{P\sim\mathcal{T}} Q(\eta_2) &\leqslant (Q'(\eta_2) + \epsilon)(1-\delta) + 13L\sigma^2\delta \\ &\leqslant \left(\frac{1}{2}\|w^*\|^2 - C + \epsilon\right)(1-\delta) + 13L\sigma^2\delta \\ &\leqslant \frac{1}{2}\|w^*\|^2 - C + 13L\sigma^2\delta + \epsilon \leqslant \frac{1}{2}\|w^*\|^2 - \frac{9C}{10},\end{aligned}$$

where the last inequality assumes $\delta \leqslant \frac{C}{260L\sigma^2}$ and $\epsilon \leqslant \frac{C}{20}$.

When $\eta \in [0, \eta_1] \cup [\eta_3, \frac{1}{c_5 d^2 \log^2(d/\epsilon)}]$, we have

$$\begin{aligned}\mathbb{E}_{P\sim\mathcal{T}} Q(\eta_2) &\geqslant (Q'(\eta) - \epsilon)(1-\delta) - 13L\sigma^2\delta \\ &\geqslant \left(\frac{1}{2}\|w^*\|^2 - \frac{C}{2} - \epsilon\right)(1-\delta) - 13L\sigma^2\delta \\ &\geqslant \frac{1}{2}\|w^*\|^2 - \frac{C}{2} - \frac{\delta}{2} - 13L\sigma^2\delta - \epsilon \geqslant \frac{1}{2}\|w^*\|^2 - \frac{6C}{10},\end{aligned}$$

where the last inequality holds as long as $\delta \leqslant \frac{C}{280L\sigma^2}$ and $\epsilon \leqslant \frac{C}{20}$.

According to Lemma D.4.8, we know $\delta \leqslant \exp(-\Omega(d))$. Therefore, the conditions for δ can be satisfied as long as d is larger than certain constant. The condition on ϵ can be satisfied as long as $\eta \leqslant \frac{1}{c_5 d^2 \log^2(d)}$ for some constant c_5. □

Generalization for $\eta \in [0, \frac{1}{c_5 d^2 \log^2 d}]$

In this section, we prove Lemma D.4.7 by showing that $\widehat{F}_{TbV}(\eta)$ is point-wise close to $F_{TbV}(\eta)$ for all $\eta \in [0, \frac{1}{c_5 d^2 \log^2(d/\epsilon)}]$. Recall Lemma D.4.7 as follows.

Lemma D.4.7. *For any $1 > \epsilon > 0$, assume σ is a constant and $d \geqslant c_4 \log(1/\epsilon)$ for some constant c_4. There exists constant c_5 such that with probability at least $1 - O(1/\epsilon)\exp(-\Omega(\epsilon^2 m))$,*

$$|\widehat{F}_{TbV}(\eta) - F_{TbV}(\eta)| \leqslant \epsilon,$$

for all $\eta \in [0, \frac{1}{c_5 d^2 \log^2(d/\epsilon)}]$.

In order to prove Lemma D.4.7, we first show that for a fixed η with high probability $\widehat{F}_{TbV}(\eta)$ is close to $F_{TbV}(\eta)$. Similar as in Lemma D.2.17, we can still show that each $\Delta_{TbV}(\eta, P)$ is $O(1)$-subexponential. The proof is deferred into Section D.4.2.

Lemma D.4.10. *Suppose* σ *is a constant. Given any* $1 > \epsilon > 0$, *for any fixed* η *with probability at least* $1 - \exp(-\Omega(\epsilon^2 m))$,

$$\left|\widehat{F}_{TbV}(\eta) - F_{TbV}(\eta)\right| \leqslant \epsilon.$$

Next, we show that there exists an ϵ-net for F_{TbV} with size $O(1/\epsilon)$. By ϵ-net, we mean there exists a finite set N_ϵ of step sizes such that $|F_{TbV}(\eta) - F_{TbV}(\eta')| \leqslant \epsilon$ for any η and $\eta' \in \arg\min_{\eta \in N_\epsilon} |\eta - \eta'|$. The proof is very similar as in Lemma D.2.18. We defer the proof of Lemma D.4.11 into Section D.4.2.

Lemma D.4.11. *Suppose* σ *is a constant. For any* $1 > \epsilon > 0$, *assume* $d \geqslant c_4 \log(1/\epsilon)$ *for some* c_4. *There exists constant* c_5 *and an* ϵ*-net* $N_\epsilon \subset [0, \frac{1}{c_5 d^2 \log^2(d/\epsilon)}]$ *for* F_{TbV} *with* $|N_\epsilon| = O(1/\epsilon)$. *That means, for any* $\eta \in [0, \frac{1}{c_5 d^2 \log^2(d/\epsilon)}]$,

$$|F_{TbV}(\eta) - F_{TbV}(\eta')| \leqslant \epsilon,$$

for $\eta' \in \arg\min_{\eta \in N_\epsilon} |\eta - \eta'|$.

Next, we show that with high probability, there also exists an ϵ-net for $\widehat{F}_{TbV}$ with size $O(1/\epsilon)$. The proof is very similar as the proof of Lemma D.2.19. We defer the proof into Section D.4.2.

Lemma D.4.12. *Suppose* σ *is a constant. For any* $1 > \epsilon > 0$, *assume* $d \geqslant c_4 \log(1/\epsilon)$ *for some* c_4. *With probability at least* $1 - \exp(-\Omega(\epsilon^2 m))$, *there exists constant* c_5 *and*

an ϵ-net $N'_\epsilon \subset [0, \frac{1}{c_5 d^2 \log^2(d/\epsilon)}]$ for $\widehat{F}_{TbV}$ with $|N_\epsilon| = O(1/\epsilon)$. That means, for any $\eta \in [0, \frac{1}{c_5 d^2 \log^2(d/\epsilon)}]$,

$$|\widehat{F}_{TbV}(\eta) - \widehat{F}_{TbV}(\eta')| \leqslant \epsilon,$$

for $\eta' \in \arg\min_{\eta \in N_\epsilon} |\eta - \eta'|$.

Combing Lemma D.4.10, Lemma D.4.11 and Lemma D.4.12, now we give the proof of Lemma D.4.7.

Proof of Lemma D.4.7. The proof is almost the same as the proof of Lemma D.2.12. We omit the details here. □

Proofs of Technical Lemmas

In Lemma D.4.13, we show when the step size is small, the expected SGD noise square is well bounded. The proof follows from the analysis in Lemma D.4.3.

Lemma D.4.13. *Let $\{w'_{\tau,\eta}\}$ be an SGD sequence running on task P without truncation. Let $n'_{\tau,\eta}$ be the SGD noise at $w'_{\tau,\eta}$. Assume $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{train}) \leqslant \sqrt{L}\sqrt{\sigma}$ for all $i \in [n]$ and $\|\xi_{train}\| \leqslant \sqrt{d}\sigma$. Suppose $\eta \in [0, \frac{1}{2L^3 d}]$, we have*

$$\mathbb{E}_{SGD}\left\|n'_{\tau,\eta}\right\|^2 \leqslant 4L^3\sigma^2 d$$

for all $\tau \leqslant t$.

Proof of Lemma D.4.13. Similar as the analysis in Lemma D.4.3, for $\eta \leqslant \frac{1}{2L^3 d}$, we have

$$\mathbb{E}_{\mathrm{SGD}}\left[\left\|n'_{\tau,\eta}\right\|^2 | w'_{\tau-1,\eta}\right] \leqslant L^2 d\left\|w'_{\tau-1,\eta} - w_{\mathrm{train}}\right\|^2.$$

and

$$\mathbb{E}_{\mathrm{SGD}}\left\|w'_{\tau-1,\eta} - w_{\mathrm{train}}\right\|^2 \leqslant (1 - \frac{\eta}{2L})^{\tau-1}\|w_{\mathrm{train}}\|^2 \leqslant \left\|w^*_{\mathrm{train}} + (X_{\mathrm{train}})^\dagger \xi_{\mathrm{train}}\right\|^2 \leqslant 4L\sigma^2.$$

Therefore, we have

$$\mathbb{E}_{\mathrm{SGD}}\left\|n'_{\tau,\eta}\right\|^2 \leqslant L^2 d\mathbb{E}_{\mathrm{SGD}}\left\|w'_{\tau,\eta} - w_{\mathrm{train}}\right\|^2 \leqslant 4L^3\sigma^2 d.$$

□

Proof of Lemma D.4.8. We can expand $Q'(\eta)$ as follows,

$$\begin{aligned}
Q'(\eta) :=& \frac{1}{2}\mathbb{E}_{\mathrm{SGD}}\left\|w'_{t,\eta} - w^*\right\|^2 \\
=& \frac{1}{2}\mathbb{E}_{\mathrm{SGD}}\left\|B_{t,\eta}w^*_{\mathrm{train}} + B_{t,\eta}(X_{\mathrm{train}})^\dagger\xi_{\mathrm{train}} - \eta\sum_{\tau=0}^{t-1}(I-\eta H_{\mathrm{train}})^{t-1-\tau}n'_{\tau,\eta} - w^*\right\|^2 \\
=& \frac{1}{2}\|B_{t,\eta}w^*_{\mathrm{train}} - w^*\|^2 + \frac{1}{2}\left\|B_{t,\eta}(X_{\mathrm{train}})^\dagger\xi_{\mathrm{train}}\right\|^2 \\
&+ \frac{\eta^2}{2}\mathbb{E}_{\mathrm{SGD}}\left\|\sum_{\tau=0}^{t-1}(I-\eta H_{\mathrm{train}})^{t-1-\tau}n'_{\tau,\eta}\right\|^2 + \left\langle B_{t,\eta}w^*_{\mathrm{train}} - w^*\right\rangle B_{t,\eta}(X_{\mathrm{train}})^\dagger\xi_{\mathrm{train}}
\end{aligned}$$

Denote

$$\begin{aligned}
G(\eta) :=& \frac{1}{2}\|B_{t,\eta}w^*_{\mathrm{train}} - w^*\|^2 + \frac{1}{2}\left\|B_{t,\eta}(X_{\mathrm{train}})^\dagger\xi_{\mathrm{train}}\right\|^2 \\
&+ \frac{\eta^2}{2}\mathbb{E}_{\mathrm{SGD}}\left\|\sum_{\tau=0}^{t-1}(I-\eta H_{\mathrm{train}})^{t-1-\tau}n'_{\tau,\eta}\right\|^2.
\end{aligned}$$

We first show that with probability at least $1-\exp(-\Omega(d))$, there exist $\eta_1,\eta_2,\eta_3 = \Theta(1/t)$ with $\eta_1 < \eta_2 < \eta_3$ such that $G(\eta_2) \leqslant 1/2\|w^*\|^2 - 5C/4$ and $G(\eta) \geqslant 1/2\|w^*\|^2 - C/4$ for all $\eta \in [0,\eta_1]\cup[\eta_3, 1/L]$.

According to Lemma D.2.3, we know with probability at least $1-\exp(-\Omega(d))$, $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\mathrm{train}}) \leqslant \sqrt{L}\sqrt{d}$ and $1/L \leqslant \lambda_i(H_{\mathrm{train}}) \leqslant L$ for all $i \in [n]$. According to Lemma D.5.1, we know with probability at least $1-\exp(-\Omega(d))$, $\sqrt{d}\sigma/4 \leqslant \|\xi_{\mathrm{train}}\| \leqslant \sqrt{d}\sigma$.

Upper bounding $G(\eta_2)$: We can expand $G(\eta)$ as follows:

$$\begin{aligned}
G(\eta) :=& \frac{1}{2}\|B_{t,\eta}w^*_{\text{train}} - w^*\|^2 + \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2 \\
&+ \frac{\eta^2}{2}\mathbb{E}_{\text{SGD}}\left\|\sum_{\tau=0}^{t-1}(I-\eta H_{\text{train}})^{t-1-\tau}n'_{\tau,\eta}\right\|^2 \\
=& \frac{1}{2}\|w^*\|^2 + \frac{1}{2}\|B_{t,\eta}w^*_{\text{train}}\|^2 + \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2 \\
&+ \frac{\eta^2}{2}\mathbb{E}_{\text{SGD}}\left\|\sum_{\tau=0}^{t-1}(I-\eta H_{\text{train}})^{t-1-\tau}n'_{\tau,\eta}\right\|^2 - \left\langle B_{t,\eta}w^*_{\text{train}}\right\rangle w^*.
\end{aligned}$$

Same as in Lemma D.2.14, we know $\frac{1}{2}\|B_{t,\eta}w^*_{\text{train}}\|^2 + \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2 \leqslant L^3\eta^2t^2\sigma^2$. For the SGD noise, by Lemma D.4.13 we know $\mathbb{E}_{\text{SGD}}\left\|n'_{\tau,\eta}\right\|^2 \leqslant 4L^3\sigma^2 d$ for all $\tau \leqslant t$ as long as $\eta \leqslant \frac{1}{2L^3 d}$. Therefore,

$$\frac{\eta^2}{2}\mathbb{E}_{\text{SGD}}\left\|\sum_{\tau=0}^{t-1}(I-\eta H_{\text{train}})^{t-1-\tau}n'_{\tau,\eta}\right\|^2 \leqslant \frac{\eta^2}{2}\sum_{\tau=0}^{t-1}\mathbb{E}_{\text{SGD}}\left\|n'_{\tau,\eta}\right\|^2 \leqslant 2L^3\eta^2\sigma^2 dt \leqslant 2L^3\eta^2\sigma^2t^2,$$

where the last inequality assumes $t \geqslant d$. According to Lemma D.2.16, for any fixed $\eta \in [0, L/t]$, with probability at least $1 - \exp(-\Omega(d))$ over X_{train},

$$\left\langle B_{t,\eta}w^*_{\text{train}}\right\rangle w^* \geqslant \frac{\eta t}{16L}.$$

Therefore, for any step size $\eta \leqslant \frac{1}{2L^3 d}$,

$$G(\eta) \leqslant \frac{1}{2}\|w^*\|^2 + 3L^3\eta^2\sigma^2t^2 - \frac{\eta t}{16L} \leqslant \frac{1}{2}\|w^*\|^2 - \frac{\eta t}{32L},$$

where the second inequality holds as long as $\eta \leqslant \frac{1}{96L^4\sigma^2 t}$. Choosing $\eta_2 := \frac{1}{96L^4\sigma^2 t}$ that is smaller than $\frac{1}{2L^3 d}$ assuming $t \geqslant d$. Then, we have

$$G(\eta_2) \leqslant \frac{1}{2}\|w^*\|^2 - \frac{5C}{4},$$

where constant $C = \frac{1}{3072L^5\sigma^2}$.

Lower bounding $G(\eta)$ for $\eta \in [0, \eta_1]$: Now, we prove that there exists $\eta_1 = \Theta(1/t)$ with $\eta_1 < \eta_2$ such that for any $\eta \in [0, \eta_1], G(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{C}{4}$. Recall that

$$\begin{aligned} G(\eta) =& \frac{1}{2}\|w^*\|^2 + \frac{1}{2}\|B_{t,\eta} w^*_{\text{train}}\|^2 + \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2 \\ &+ \frac{\eta^2}{2}\mathbb{E}_{\text{SGD}}\left\|\sum_{\tau=0}^{t-1}(I - \eta H_{\text{train}})^{t-1-\tau} n'_{\tau,\eta}\right\|^2 - \langle B_{t,\eta} w^*_{\text{train}}\rangle w^*. \\ \geqslant& \frac{1}{2}\|w^*\|^2 - \langle B_{t,\eta} w^*_{\text{train}}\rangle w^*. \end{aligned}$$

Same as in Lemma D.2.14, by choosing $\eta_1 = \frac{C}{4Lt}$, we have for any $\eta \in [0, \eta_1]$,

$$G(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{C}{4}.$$

Lower bounding $G(\eta)$ for $\eta \in [\eta_3, 1/L]$: Now, we prove that there exists $\eta_3 = \Theta(1/t)$ with $\eta_3 > \eta_2$ such that for all $\eta \in [\eta_3, 1/L]$,

$$G(\eta) \geqslant \frac{1}{2}\|w^*\|^2 - \frac{C}{4}.$$

Recall that

$$\begin{aligned} G(\eta) =& \frac{1}{2}\|B_{t,\eta} w^*_{\text{train}} - w^*\|^2 + \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2 \\ &+ \frac{\eta^2}{2}\mathbb{E}_{\text{SGD}}\left\|\sum_{\tau=0}^{t-1}(I - \eta H_{\text{train}})^{t-1-\tau} n'_{\tau,\eta}\right\|^2 \\ \geqslant& \frac{1}{2}\left\|B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right\|^2. \end{aligned}$$

Same as in Lemma D.2.14, by choosing $\eta_3 = \log(2)L/t$, as long as $\sigma \geqslant 8\sqrt{L}$, we have

$$G(\eta) \geqslant \frac{1}{2}\|w^*\|^2$$

for all $\eta \in [\eta_3, 1/L]$. Note $\eta_3 \leqslant 1/L$ as long as $t \geqslant \log(2)L^2$.

Overall, we have shown that there exist $\eta_1, \eta_2, \eta_3 = \Theta(1/t)$ with $\eta_1 < \eta_2 < \eta_3$ such that $G(\eta_2) \leqslant 1/2\|w^*\|^2 - 5C/4$ and $G(\eta) \geqslant 1/2\|w^*\|^2 - C/4$ for all $\eta \in [0, \eta_1] \cup [\eta_3, 1/L]$. Recall that $Q'(\eta) = G(\eta) + \langle B_{t,\eta} w^*_{\text{train}} - w^* \rangle B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}$. Choosing $\epsilon = C/4$ in Lemma D.2.15, we know with probability at least $1 - \exp(-\Omega(d))$, $\left|\langle B_{t,\eta} w^*_{\text{train}} - w^* \rangle B_{t,\eta}(X_{\text{train}})^\dagger \xi_{\text{train}}\right| \leqslant C/4$ for all $\eta \in [0, 1/L]$. Therefore, we know $Q'(\eta_2) \leqslant 1/2\|w^*\|^2 - C$ and $Q'(\eta) \geqslant 1/2\|w^*\|^2 - C/2$ for all $\eta \in [0, \eta_1] \cup [\eta_3, 1/L]$. □

In order to prove Lemma D.4.9, we first construct a super-martingale to show that as long as task P is well behaved, with high probability in SGD noise, the weight norm along the trajectory never exceeds $4\sqrt{L}\sigma$.

Lemma D.4.14. *Assume $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{train}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{train}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{train}\| \leqslant \sqrt{d}\sigma$. Given any $1 > \delta > 0$, suppose $\eta \leqslant \frac{1}{c_5 d^2 \log^2(d/\delta)}$ for some constant c_5, with probability at least $1 - \delta$ in the SGD noise,*

$$\left\|w'_{\tau,\eta}\right\| < 4\sqrt{L}\sigma$$

for all $\tau \leqslant t$.

Proof of Lemma D.4.14. According to the proofs of Lemma D.4.13, as long as $\eta \leqslant \frac{1}{2L^3 d}$, we have

$$\mathbb{E}_{\text{SGD}}\left[\left\|w'_{t,\eta} - w_{\text{train}}\right\|^2 | w'_{t-1,\eta}\right] \leqslant (1 - \frac{\eta}{2L})\left\|w'_{t-1,\eta} - w_{\text{train}}\right\|^2.$$

Since log is a concave function, by Jenson's inequality, we know

$$\begin{aligned}&\mathbb{E}_{\mathrm{SGD}}\left[\log\left\|w'_{t,\eta}-w_{\mathrm{train}}\right\|^2|w'_{t-1,\eta}\right]\\&\leqslant\log\mathbb{E}_{\mathrm{SGD}}\left[\left\|w'_{t,\eta}-w_{\mathrm{train}}\right\|^2|w'_{t-1,\eta}\right]\leqslant\log\left\|w'_{t-1,\eta}-w_{\mathrm{train}}\right\|^2+\log\Big(1-\frac{\eta}{2L}\Big).\end{aligned}$$

Defining $G_t=\log\left\|w'_{t,\eta}-w_{\mathrm{train}}\right\|^2-t\log\big(1-\frac{\eta}{2L}\big)$, we know G_t is a super-martingale. Next, we bound the martingale differences.

We can bound $|G_t-\mathbb{E}_{\mathrm{SGD}}[G_t|w'_{t-1,\eta}]|$ as follows,

$$|G_t-\mathbb{E}_{\mathrm{SGD}}[G_t|w'_{t-1,\eta}]|\leqslant\max_{n'_{t-1,\eta},n''_{t-1,\eta}}\log\left(\frac{\left\|(I-\eta H_{\mathrm{train}})(w'_{t-1,\eta}-w_{\mathrm{train}})-\eta n'_{t-1,\eta}\right\|^2}{\left\|(I-\eta H_{\mathrm{train}})(w'_{t-1,\eta}-w_{\mathrm{train}})-\eta n''_{t-1,\eta}\right\|^2}\right)$$

We can expand $\left\|(I-\eta H_{\mathrm{train}})(w'_{t-1,\eta}-w_{\mathrm{train}})-\eta n'_{t-1,\eta}\right\|^2$ as follows,

$$\begin{aligned}&\left\|(I-\eta H_{\mathrm{train}})(w'_{t-1,\eta}-w_{\mathrm{train}})-\eta n'_{t-1,\eta}\right\|^2\\=&\left\|(I-\eta H_{\mathrm{train}})(w'_{t-1,\eta}-w_{\mathrm{train}})\right\|^2-2\eta\left\langle n'_{t-1,\eta}\right\rangle(I-\eta H_{\mathrm{train}})(w'_{t-1,\eta}-w_{\mathrm{train}})\\&+\eta^2\left\|n'_{t-1,\eta}\right\|^2\end{aligned}$$

We can bound the norm of the noise as follows,

$$\begin{aligned}\left\|n'_{t-1,\eta}\right\|&=\left\|x_{i(t-1)}x_{i(t-1)}^\top(w'_{t-1,\eta}-w_{\mathrm{train}})-H_{\mathrm{train}}(w'_{t-1,\eta}-w_{\mathrm{train}})\right\|\\&\leqslant\left\|x_{i(t-1)}x_{i(t-1)}^\top(w'_{t-1,\eta}-w_{\mathrm{train}})\right\|+\left\|H_{\mathrm{train}}(w'_{t-1,\eta}-w_{\mathrm{train}})\right\|\\&\leqslant(Ld+L)\left\|w'_{t-1,\eta}-w_{\mathrm{train}}\right\|\leqslant 2Ld\left\|w'_{t-1,\eta}-w_{\mathrm{train}}\right\|,\end{aligned}$$

where the second inequality uses $\left\|x_{i(t-1)}\right\|\leqslant\sqrt{Ld}$. Therefore, we have

$$\begin{aligned}&\left|2\eta\left\langle n'_{t-1,\eta}\right\rangle(I-\eta H_{\mathrm{train}})(w'_{t-1,\eta}-w_{\mathrm{train}})\right|\leqslant 4L\eta d\left\|w'_{t-1,\eta}-w_{\mathrm{train}}\right\|^2,\\&\eta^2\left\|n'_{t-1,\eta}\right\|^2\leqslant 4L^2\eta^2d^2\left\|w'_{t-1,\eta}-w_{\mathrm{train}}\right\|^2.\end{aligned}$$

This further implies,

$$
\begin{aligned}
&|G_t - \mathbb{E}_{\mathrm{SGD}}[G_t|w'_{t-1,\eta}]| \\
\leqslant &\log\left(\frac{\left\|(I-\eta H_{\mathrm{train}})(w'_{t-1,\eta}-w_{\mathrm{train}})\right\|^2 + (4L\eta d + 4L^2\eta^2 d^2)\left\|w'_{t-1,\eta}-w_{\mathrm{train}}\right\|^2}{\left\|(I-\eta H_{\mathrm{train}})(w'_{t-1,\eta}-w_{\mathrm{train}})\right\|^2 - 4L\eta d\left\|w'_{t-1,\eta}-w_{\mathrm{train}}\right\|^2}\right) \\
\leqslant &\log\left(1+\frac{8L\eta d + 4L^2\eta^2 d^2}{(1-2L\eta-4L\eta d)}\right) \leqslant 16L\eta d + 8L^2\eta^2 d^2,
\end{aligned}
$$

where the second inequality uses the fact that $\left\|(I-\eta H_{\mathrm{train}})(w'_{t-1,\eta}-w_{\mathrm{train}})\right\|^2 \geqslant (1-2L\eta)\left\|w'_{t-1,\eta}-w_{\mathrm{train}}\right\|^2$. The last inequality assumes $\eta \leqslant \frac{1}{12Ld}$ and uses numerical inequality $\log(1+x) \leqslant x$. Assuming $\eta \leqslant 1/(Ld)$, we further have $|G_t - \mathbb{E}_{\mathrm{SGD}}[G_t|w'_{t-1,\eta}]| \leqslant L^2\eta d$.

By Azuma's inequality, we know with probability at least $1-\delta/t$,

$$
G_t \leqslant G_0 + L^2\sqrt{2t}\eta d\log(t/\delta).
$$

Plugging in $G_t = \log\left\|w'_{t,\eta}-w_{\mathrm{train}}\right\|^2 - t\log\left(1-\frac{\eta}{2L}\right)$ and $G_0 = \log\|w_0-w_{\mathrm{train}}\|^2 = \log\|w_{\mathrm{train}}\|^2$, we have

$$
\begin{aligned}
\log\left\|w'_{t,\eta}-w_{\mathrm{train}}\right\|^2 \leqslant &\log\|w_{\mathrm{train}}\|^2 + t\log\left(1-\frac{\eta}{2L}\right) + L^2\sqrt{2t}\eta d\log(t/\delta) \\
\leqslant &\log\|w_{\mathrm{train}}\|^2 - \frac{\eta}{2L}t + L^2\sqrt{2t}\eta d\log(t/\delta).
\end{aligned}
$$

This implies,

$$
\begin{aligned}
\left\|w'_{t,\eta}-w_{\mathrm{train}}\right\|^2 \leqslant &\|w_{\mathrm{train}}\|^2 \exp\left(\eta\left(-\frac{1}{2L}t + L^2\sqrt{2}\log(t/\delta)d\sqrt{t}\right)\right) \\
= &\|w_{\mathrm{train}}\|^2 \exp\left(O(d^2\log^2(d/\delta))\eta\right) \\
\leqslant &\|w_{\mathrm{train}}\|^2 \exp\left(2/3\right),
\end{aligned}
$$

where the second inequality assumes $\eta \leqslant \frac{1}{c_5 d^2 log^2(d/\delta)}$ for some constant c_5. Further-

more, since $\|w_{\text{train}}\| \leqslant (1+\sqrt{L})\sigma$, we have $\left\|w'_{t,\eta}\right\| \leqslant (1+e^{1/3})\|w_{\text{train}}\| < 4\sqrt{L}\sigma$.

Overall, we know as long as $\eta \leqslant \frac{1}{c_5 d^2 log^2(d/\delta)}$, with probability at least $1-\delta/t$, $\left\|w'_{t,\eta}\right\| \leqslant 4\sqrt{L}\sigma$. Since this analysis also applies to any $\tau \leqslant t$, we know for any τ, with probability at least $1-\delta/t$, $\left\|w'_{\tau,\eta}\right\| < 4\sqrt{L}\sigma$. Taking a union bound over $\tau \leqslant t$, we have with probability at least $1-\delta$, $\left\|w'_{\tau,\eta}\right\| < 4\sqrt{L}\sigma$ for all $\tau \leqslant t$. □

Proof of Lemma D.4.9. Let $\mathcal{E}$ be the event that $\left\|w'_{\tau,\eta}\right\| < 4\sqrt{L}\sigma$ for all $\tau \leqslant t$. We first show that $\mathbb{E}_{\text{SGD}}\|w_{t,\eta} - w^*\|^2$ is close to $\mathbb{E}_{\text{SGD}}\left\|w'_{t,\eta} - w^*\right\|^2 \mathbb{1}\{\mathcal{E}\}$. It's not hard to verify that

$$\mathbb{E}_{\text{SGD}}\|w_{t,\eta} - w^*\|^2 = \mathbb{E}_{\text{SGD}}\left\|w'_{t,\eta} - w^*\right\|^2 \mathbb{1}\{\mathcal{E}\} + \|u - w^*\|^2 \Pr\left[\bar{\mathcal{E}}\right],$$

where u is a fixed vector with norm $4\sqrt{L}\sigma$. By Lemma D.4.14, we know $\Pr\left[\bar{\mathcal{E}}\right] \leqslant \epsilon/(25L\sigma^2)$ as long as $\eta \leqslant \frac{1}{c_5 d^2 \log^2(d/\epsilon)}$ for some constant c_5. Therefore, we have

$$\left|\mathbb{E}_{\text{SGD}}\|w_{t,\eta} - w^*\|^2 - \mathbb{E}_{\text{SGD}}\left\|w'_{t,\eta} - w^*\right\|^2 \mathbb{1}\{\mathcal{E}\}\right| \leqslant \epsilon.$$

Next, we show that $\mathbb{E}_{\text{SGD}}\left\|w'_{t,\eta} - w^*\right\|^2 \mathbb{1}\{\mathcal{E}\}$ is close to $\mathbb{E}_{\text{SGD}}\left\|w'_{t,\eta} - w^*\right\|^2$. For any $1 \leqslant \tau \leqslant t$, let $\mathcal{E}_\tau$ be the event that $\left\|w'_{\tau,\eta}\right\| \geqslant 4\sqrt{L}\sigma$ and $\left\|w'_{\tau',\eta}\right\| < 4\sqrt{L}\sigma$ for all $\tau' < \tau$. Basically $\mathcal{E}_\tau$ means the weight norm exceeds the threshold at step τ for the first time. It's easy to see that $\cup_{\tau=1}^{t}\mathcal{E}_\tau = \bar{\mathcal{E}}$. Therefore, we have

$$\mathbb{E}_{\text{SGD}}\left\|w'_{t,\eta} - w^*\right\|^2 = \mathbb{E}_{\text{SGD}}\left\|w'_{t,\eta} - w^*\right\|^2 \mathbb{1}\{\mathcal{E}\} + \sum_{\tau=1}^{t} \mathbb{E}_{\text{SGD}}\left\|w'_{t,\eta} - w^*\right\|^2 \mathbb{1}\{\mathcal{E}_\tau\}.$$

Conditioning on $\mathcal{E}_\tau$, we know $\left\|w'_{\tau-1,\eta}\right\| < 4\sqrt{L}\sigma$. Since we assume $\frac{\sqrt{d}}{\sqrt{L}} \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{L}\sqrt{d}$ for all $i \in [n]$ and $\xi_{\text{train}} \leqslant \sqrt{d}\sigma$, we know $\|w_{\text{train}}\| \leqslant 2\sqrt{L}\sigma$. Therefore, we have

$\left\|w'_{\tau-1,\eta} - w_{\text{train}}\right\| \leqslant 6\sqrt{L}\sigma$. Recall the SGD updates,

$$w'_{\tau,\eta} - w_{\text{train}} = (I - \eta H_{\text{train}})(w'_{\tau-1,\eta} - w_{\text{train}}) - \eta n'_{\tau-1,\eta}.$$

For the noise term, we have $\eta\left\|n'_{\tau-1,\eta}\right\| \leqslant 2\eta L d\left\|w'_{\tau-1,\eta} - w_{\text{train}}\right\|$ that is at most $\left\|w'_{\tau-1,\eta} - w_{\text{train}}\right\|$ assuming $\eta \leqslant \frac{1}{2Ld}$. Therefore, $\left\|w'_{\tau,\eta} - w_{\text{train}}\right\| \leqslant 2\left\|w'_{\tau-1,\eta} - w_{\text{train}}\right\| \leqslant 12\sqrt{L}\sigma$. Note that event $\mathcal{E}_\tau$ is independent with the SGD noises after step τ. Therefore, according to the previous analysis, we know as long as $\eta \leqslant \frac{1}{2L^3 d}$,

$$\mathbb{E}_{\text{SGD}}\left[\left\|w'_{t,\eta} - w_{\text{train}}\right\|^2 | \mathcal{E}_\tau\right] \leqslant \left\|w'_{\tau,\eta} - w_{\text{train}}\right\|^2 \leqslant 2L^2\sigma^2.$$

Then, we can bound $\mathbb{E}_{\text{SGD}}\left[\left\|w'_{t,\eta} - w^*\right\|^2 | \mathcal{E}_\tau\right]$ as follows,

$$\begin{aligned}
&\mathbb{E}_{\text{SGD}}\left[\left\|w'_{t,\eta} - w^*\right\|^2 | \mathcal{E}_\tau\right] \\
=&\mathbb{E}_{\text{SGD}}\left[\left\|w'_{t,\eta} - w_{\text{train}} + w_{\text{train}} - w^*\right\|^2 | \mathcal{E}_\tau\right] \\
\leqslant&\mathbb{E}_{\text{SGD}}\left[\left\|w'_{t,\eta} - w_{\text{train}}\right\|^2 | \mathcal{E}_\tau\right] + 2\mathbb{E}_{\text{SGD}}\left[\left\|w'_{t,\eta} - w_{\text{train}}\right\| | \mathcal{E}_\tau\right]\left\|w_{\text{train}} - w^*\right\| \\
&+ \left\|w_{\text{train}} - w^*\right\|^2 \\
\leqslant&2L^2\sigma^2 + 2 \cdot 2L\sigma \cdot 3\sqrt{L}\sigma + 9L\sigma^2 \leqslant 3L^2\sigma^2.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\sum_{\tau=1}^{t} \mathbb{E}_{\text{SGD}}\left\|w'_{t,\eta} - w^*\right\|^2 \mathbb{1}\left\{\mathcal{E}_\tau\right\} =& \sum_{\tau=1}^{t} \mathbb{E}_{\text{SGD}}\left[\left\|w'_{t,\eta} - w^*\right\|^2 | \mathcal{E}_\tau\right] \Pr[\mathcal{E}_\tau] \\
\leqslant& 3L^2\sigma^2 \sum_{\tau=1}^{t} \Pr[\mathcal{E}_\tau] = 3L^2\sigma^2 \Pr\left[\bar{\mathcal{E}}\right] \leqslant 3L^2\sigma^2\epsilon.
\end{aligned}$$

This then implies that $\left|\mathbb{E}_{\text{SGD}}\left\|w'_{t,\eta} - w^*\right\|^2 - \mathbb{E}_{\text{SGD}}\left\|w'_{t,\eta} - w^*\right\|^2 \mathbb{1}\left\{\mathcal{E}\right\}\right| \leqslant 3L^2\sigma^2\epsilon$.

Finally, we have

$$\begin{aligned}
&\left|\mathbb{E}_{\mathrm{SGD}}\|w_{t,\eta}-w^*\|^2-\mathbb{E}_{\mathrm{SGD}}\left\|w'_{t,\eta}-w^*\right\|^2\right| \\
\leqslant&\left|\mathbb{E}_{\mathrm{SGD}}\|w_{t,\eta}-w^*\|^2-\mathbb{E}_{\mathrm{SGD}}\left\|w'_{t,\eta}-w^*\right\|^2\mathbb{1}\left\{\mathcal{E}\right\}\right| \\
&+\left|\mathbb{E}_{\mathrm{SGD}}\left\|w'_{t,\eta}-w^*\right\|^2-\mathbb{E}_{\mathrm{SGD}}\left\|w'_{t,\eta}-w^*\right\|^2\mathbb{1}\left\{\mathcal{E}\right\}\right| \\
\leqslant&\left(3L^2\sigma^2+1\right)\epsilon
\end{aligned}$$

as long as $\eta \leqslant \frac{1}{c_5 d^2 \log^2(d/\epsilon)}$. Therefore, $|Q(\eta)-Q'(\eta)| \leqslant (3L^2\sigma^2+1)\,\epsilon/2$. Choosing $\epsilon' = \frac{2\epsilon}{(3L^2\sigma^2+1)}$ finishes the proof. □

Proof of Lemma D.4.10. Recall that

$$\widehat{F}_{TbV}(\eta) := \frac{1}{m}\sum_{k=1}^{m}\Delta_{TbV}(\eta,P) = \frac{1}{m}\sum_{k=1}^{m}\mathbb{E}_{\mathrm{SGD}}\frac{1}{2}\left\|w_{t,\eta}^{(k)}-w_{\mathrm{valid}}^{(k)}\right\|_{H_{\mathrm{valid}}^{(k)}}^2.$$

Similar as in Lemma D.2.12, we can show $\frac{1}{2}\left\|w_{t,\eta}^{(k)}-w_{\mathrm{valid}}^{(k)}\right\|_{H_{\mathrm{valid}}^{(k)}}^2$ is $O(1)$-subexponential, which implies

$\mathbb{E}_{\mathrm{SGD}}\frac{1}{2}\left\|w_{t,\eta}^{(k)}-w_{\mathrm{valid}}^{(k)}\right\|_{H_{\mathrm{valid}}^{(k)}}^2$ is $O(1)$-subexponential. Therefore, $\widehat{F}_{TbV}(\eta)$ is the average of m i.i.d. $O(1)$-subexponential random variables. By standard concentration inequality, we know for any $1>\epsilon>0$, with probability at least $1-\exp(-\Omega(\epsilon^2 m))$,

$$\left|\widehat{F}_{TbV}(\eta)-F_{TbV}(\eta)\right| \leqslant \epsilon.$$

□

Proof of Lemma D.4.11. Recall that

$$F_{TbV}(\eta) = \mathbb{E}_{P\sim\mathcal{T}}\mathbb{E}_{\mathrm{SGD}}\frac{1}{2}\|w_{t,\eta}-w^*\|^2+\sigma^2/2$$

We only need to construct an ϵ-net for $\mathbb{E}_{P\sim\mathcal{T}}\mathbb{E}_{\mathrm{SGD}}\frac{1}{2}\|w_{t,\eta}-w^*\|^2$. Let $\mathcal{E}$ be the event

that $\sqrt{d}/\sqrt{L} \leqslant \sigma_i(X_{\text{train}}) \leqslant \sqrt{Ld}$ and $1/L \leqslant \lambda_i(H_{\text{train}}) \leqslant L$ for all $i \in [n]$ and $\sqrt{d}\sigma/4 \leqslant \|\xi_{\text{train}}\| \leqslant \sqrt{d}\sigma$ We have

$$\begin{aligned}
&\mathbb{E}_{P\sim\mathcal{T}}\mathbb{E}_{\text{SGD}}\frac{1}{2}\|w_{t,\eta} - w^*\|^2 \\
=&\mathbb{E}_{P\sim\mathcal{T}}\left[\frac{1}{2}\mathbb{E}_{\text{SGD}}\|w_{t,\eta} - w^*\|^2|\mathcal{E}\right]\Pr[\mathcal{E}] + \mathbb{E}_{P\sim\mathcal{T}}\left[\frac{1}{2}\mathbb{E}_{\text{SGD}}\|w_{t,\eta} - w^*\|^2|\bar{\mathcal{E}}\right]\Pr\big[\bar{\mathcal{E}}\big]
\end{aligned}$$

According to Lemma D.4.9, we know conditioning on $\mathcal{E}$,

$$\left|\frac{1}{2}\mathbb{E}_{\text{SGD}}\|w_{t,\eta} - w^*\|^2 - \frac{1}{2}\mathbb{E}_{\text{SGD}}\big\|w'_{t,\eta} - w^*\big\|^2\right| \leqslant \epsilon,$$

as long as $\eta \leqslant \frac{1}{c_5 d^2\log^2(d/\epsilon)}$. Note $\{w'_{\tau,\eta}\}$ is the SGD sequence without truncation.

For the second term, we have

$$\mathbb{E}_{P\sim\mathcal{T}}\left[\frac{1}{2}\mathbb{E}_{\text{SGD}}\|w_{t,\eta} - w^*\|^2|\bar{\mathcal{E}}\right]\Pr\big[\bar{\mathcal{E}}\big] \leqslant 13L\sigma^2\Pr\big[\bar{\mathcal{E}}\big] \leqslant \epsilon,$$

where the last inequality assumes $\Pr\big[\bar{\mathcal{E}}\big] \leqslant \frac{\epsilon}{13L\sigma^2}$. According to Lemma D.2.3 and Lemma D.5.1, we know $\Pr\big[\bar{\mathcal{E}}\big] \leqslant \exp(-\Omega(d))$. Therefore, given any $\epsilon > 0$, we have $\Pr\big[\bar{\mathcal{E}}\big] \leqslant \frac{\epsilon}{13L\sigma^2}$ as long as $d \geqslant c_4\log(1/\epsilon)$ for some constant c_4.

Then, we only need to construct an ϵ-net for $\mathbb{E}_{P\sim\mathcal{T}}\left[\frac{1}{2}\mathbb{E}_{\text{SGD}}\big\|w'_{t,\eta} - w^*\big\|^2|\mathcal{E}\right]\Pr[\mathcal{E}]$. By the analysis in Lemma D.4.3, it's not hard to prove

$$\left|\frac{\partial}{\partial\eta}\mathbb{E}_{P\sim\mathcal{T}}\left[\frac{1}{2}\mathbb{E}_{\text{SGD}}\big\|w'_{t,\eta} - w^*\big\|^2|\mathcal{E}\right]\Pr[\mathcal{E}]\right| = O(1)t(1 - \frac{\eta}{2L})^{t-1},$$

for all $\eta \in [0, \frac{1}{c_5 d^2\log^2(d/\epsilon)}]$. Similar as in Lemma D.2.15, for any $\epsilon > 0$, we know there exists an ϵ-net N_ϵ with size $O(1/\epsilon)$ such that for any $\eta \in [0, \frac{1}{c_5 d^2\log^2(d/\epsilon)}]$,

$$\left|\mathbb{E}_{P\sim\mathcal{T}}\left[\frac{1}{2}\mathbb{E}_{\text{SGD}}\big\|w'_{t,\eta} - w^*\big\|^2|\mathcal{E}\right]\Pr[\mathcal{E}] - \mathbb{E}_{P\sim\mathcal{T}}\left[\frac{1}{2}\mathbb{E}_{\text{SGD}}\big\|w'_{t,\eta'} - w^*\big\|^2|\mathcal{E}\right]\Pr[\mathcal{E}]\right| \leqslant \epsilon$$

for $\eta' \in \arg\min_{\eta \in N_\epsilon} |\eta - \eta'|$.

Combing with the bounds on $\left|\frac{1}{2}\mathbb{E}_{\mathrm{SGD}}\|w_{t,\eta} - w^*\|^2 \mathbb{1}\left\{\mathcal{E}\right\} - \frac{1}{2}\mathbb{E}_{\mathrm{SGD}}\left\|w'_{t,\eta} - w^*\right\|^2 \mathbb{1}\left\{\mathcal{E}\right\}\right|$ and

$\mathbb{E}_{P\sim\mathcal{T}}\left[\frac{1}{2}\mathbb{E}_{\mathrm{SGD}}\|w_{t,\eta} - w^*\|^2 | \bar{\mathcal{E}}\right]\Pr\left[\bar{\mathcal{E}}\right]$, we have for any $\eta \in [0, \frac{1}{c_5 d^2 \log^2(d/\epsilon)}]$,

$$F_{TbV}(\eta) - F_{TbV}(\eta') \leqslant 4\epsilon$$

for $\eta' \in \arg\min_{\eta \in N_\epsilon} |\eta - \eta'|$. We finish the proof by replacing 4ϵ by ϵ'. □

Proof of Lemma D.4.12. The proof is very similar as the proof of Lemma D.2.19. The only difference is that we need to first relate the SGD sequence with truncation to the SGD sequence without truncation and then bound the Lipschitzness on the SGD sequence without truncation (as we did in Lemma D.4.11). We omit the details here. □

D.5 Tools

D.5.1 Norm of Random Vectors

We use the following lemma to bound the noise in least squares model.

Lemma D.5.1 (Theorem 3.1.1 in Vershynin (2018b))**.** *Let $X = (X_1, X_2, \cdots, X_n) \in \mathbb{R}^n$ be a random vector with each entry independently sampled from $\mathcal{N}(0,1)$. Then*

$$\Pr\left[\left|\|x\| - \sqrt{n}\right| \geqslant t\right] \leqslant 2\exp\left(-t^2/C^2\right),$$

where C is an absolute constant.

D.5.2 Singular Values of Gaussian Matrices

Given a random Gaussian matrix, in expectation its smallest and largest singular value can be bounded as follows.

Lemma D.5.2 (Theorem 5.32 in Vershynin (2010)). *Let A be an $N \times n$ matrix whose entries are independent standard normal random variables. Then*

$$\sqrt{N} - \sqrt{n} \leqslant \mathbb{E}s_{\min}(A) \leqslant \mathbb{E}s_{\max}(A) \leqslant \sqrt{N} + \sqrt{n}$$

Lemma D.5.3 shows a lipchitz function over i.i.d. Gaussian variables concentrate well on its mean. We use this lemma to argue for any fixed step size, the empirical meta objective concentrates on the population meta objective.

Lemma D.5.3 (Proposition 5.34 in Vershynin (2010)). *Let f be a real valued Lipschitz function on $\mathbb{R}^n$ with Lipschitz constant K. Let X be the standard normal random vector in $\mathbb{R}^n$. Then for every $t \geqslant 0$ one has*

$$\Pr[f(X) - \mathbb{E}f(X) \geqslant t] \leqslant \exp\left(-\frac{t^2}{2K^2}\right).$$

The following lemma shows a tall random Gaussian matrix is well-conditioned with high probability. The proof follows from Lemma D.5.2 and Lemma D.5.3. We use Lemma D.5.4 to show the covariance matrix is well conditioned in the least squares model.

Lemma D.5.4 (Corollary 5.35 in Vershynin (2010)). *Let A be an $N \times n$ matrix whose entries are independent standard normal random variables. Then for every $t \geqslant 0$ with probability at least $1 - 2\exp(-t^2/2)$ one has*

$$\sqrt{N} - \sqrt{n} - t \leqslant s_{\min}(A) \leqslant s_{\max}(A) \leqslant \sqrt{N} + \sqrt{n} + t$$

D.5.3 Johnson-Lindenstrauss Lemma

We also use Johnson-Lindenstrauss (JL) Lemma in some of the lemmas. JL Lemma tells us the projection of a fixed vector on a random subspace concentrates well as

long as the subspace is reasonably large.

Lemma D.5.5 (Johnson and Lindenstrauss (1984))**.** *Let P be a projection in $\mathbb{R}^d$ onto a random n-dimensional subspace uniformly distributed in $G_{d,n}$. Let $z \in \mathbb{R}^d$ be a fixed point and $\epsilon > 0$, then with probability at least $1 - 2\exp(-c\epsilon^2 n)$,*

$$(1-\epsilon)\sqrt{\frac{n}{d}}\,\|z\| \leqslant \|Pz\| \leqslant (1+\epsilon)\sqrt{\frac{n}{d}}\,\|z\|\,.$$

D.6 Experiment Details

We describe the detailed settings of our experiments in Section D.6.1 and give more experimental results in Section D.6.2.

D.6.1 Experiment Settings

Optimizing step size for quadratic objective In this experiment, we meta-train a learning rate for gradient descent on a fixed quadratic objective. Our goal is to show that the autograd module in popular deep learning softwares, such as Tensorflow, can have numerical issues when using the log-transformed meta objective. Therefore, we first implement the meta-training process with Tensorflow to see the results. We then re-implement the meta-training using the hand-derived meta-gradient (see Eqn D.1) to compare the result.

A general setting for both implementations is as follows. The inner problem is fixed as a 20-dimensional quadratic objective as described in Section 5.3, and we use the log-transformed meta objective for training. The positive semi-definite matrix H is generated by first sampling a 20×20 matrix X with all entries drawn from the standard normal distribution and then setting $H = X^T X$. The initial point w_0 is drawn from standard normal as well. Note that we use the same quadratic problem

(i.e., the same H and w_0) throughout the meta-training. We do 1000 meta-training iterations, and collect results for different settings of the initial learning rate η_0 and the unroll length t.

We first implement the meta-training code with Tensorflow. Our code is adapted from Wichrowska et al. (2017) [1]. We use their global learning rate optimizer and specify the problem set to have only one quadratic objective instance. We implemented the quadratic objective class ourselves (the "MyQuadratic" class). We also turned off multiple advanced features in the original code, such as attention and second derivatives, by assigning their flags as false. This ensures that the experiments have exactly the same settings as we described. The meta-training learning rate is set to be 0.001, which is of similar scale as our next experiment. We also try RMSProp as the meta optimizer, which alleviates some of the numerical issues as it renormalizes the gradient, but our experiments show that even RMSProp is still much worse than our implementation.

We then implement the meta-training by hand to show the accurate training results that avoid numerical issues. Specifically, we compute the meta-gradient using Eq (D.1), where we also scaled the numerator and denominator as described in Claim D.1.2 to avoid numerical issues. We use the algorithm suggested in Theorem 5.3.2, except we choose the meta-step size to be $1/(100\sqrt{k})$ as the constants in Theorem 5.3.2 were not optimized.

Train-by-train vs. train-by-validation, synthetic data In this experiment, we find the optimal learning rate η^* for least-squares problems trained in train-by-train and train-by-validation settings and then see how the learning rate works on new tasks.

[1] Their open source code is available at `https://github.com/tensorflow/models/tree/master/research/learned_optimizer`

Specifically, we generate 300 different 1000-dimensional least-squares tasks with noise as defined in Section 5.4 for inner-training and then use the meta-objectives defined in Eq (5.3) and (5.4) to find the optimal learning rate. The inner-training number of steps t is set as 40. We try different sample sizes and different noise levels for comparison. Subsequently, in order to test how the two η^* (for train-by-train and train-by-validation respectively) work, we use them on 10 test tasks (the same setting as the inner-training problem) and compute training and testing root mean squared error (RMSE).

Note that since we only need the final optimal η^* found under the two meta-objective settings (regardless of how we find it), we do not need to actually do the meta-training. Instead, we do a grid search on the interval $[10^{-6}, 1]$, which is divided log-linearly to 25 candidate points. For both the train-by-train and train-by-validation settings, we average the meta-objectives over the 300 inner problems and see which η minimizes this averaged meta-objective. The results are shown in Appendix D.6.2.

Train-by-train vs. train-by-validation, MLP optimizer on MNIST To observe the trade-off between train-by-train and train-by-validation in a broader and more realistic case, we also do experiments to meta-train an MLP optimizer as in Metz et al. (2019) to solve the MNIST classification problem. We use part of their code [2] to integrate with our code in the first experiment, and we use exactly the same default setting as theirs, which is summarized below.

The MLP optimizer is a trainable optimizer that works on each parameter separately. When doing inner-training, for each parameter, we first compute some statis-

[2] Their code is available at `https://github.com/google-research/google-research/tree/master/task_specific_learned_opt`

tics of that parameter (explained below), which are combined into a feature vector, and then feed that feature vector to a Muti-Layer Perceptron (MLP) with ReLU activations, which outputs two scalars, the update direction and magnitude. The update is computed as the direction times the exponential of the magnitude. The feature vector is 31-dimensional, which includes gradient, parameter value, first-order moving averages (5-dim), second-order moving averages (5-dim), normalized gradient (5-dim), reciprocal of square root second-order moving averages (5-dim) and a step embedding (9-dim). All moving averages are computed using 5 different decay rates (0.5, 0.9, 0.99, 0.999, 0.9999), and the step embedding is tanh distortion of the current number of steps divided by 9 different scales (3, 10, 30, 100, 300, 1000, 3000, 10000, 300000). After expanding the 31-dimensional feature vector for each parameter, we also normalize the set of vectors dimension-wise across all the parameters to have mean 0 and standard deviation 1 (except for the step embedding part). More details can be found in their original paper and original implementation.

The inner-training problem is defined as using a two-layer fully connected network (i.e., another "MLP") with ReLU activations to solve the classic MNIST 10-class classification problem. We use a very small network for computational efficiency, and the two layers have 100 and 20 neurons. We fix the cross-entropy loss as the inner-objective and use mini-batches of 32 samples when inner-training.

When we meta-train the MLP optimizer, we use exactly the same process as fixed in experiments by Wichrowska et al. (2017). We use 100 different inner problems by shuffling the 10 classes and also sampling a new subset of data if we do not use the complete MNIST data set. We run each of the problems with three inner-training trajectories starting with different initialization. Each inner-training trajectory is divided into a certain number of unrolled segments, where we compute

the meta-objective and update the meta-optimizer after each segment. The number of unrolled segments in each trajectory is sampled from $10 + \text{Exp}(30)$, and the length of each segment is sampled from $50 + \text{Exp}(100)$, where $\text{Exp}(\cdot)$ denotes the exponential distribution. Note that the meta-objective computed after each segment is defined as the average of all the inner-objectives (evaluated on the train/validation set for train-by-train/train-by-val) within that segment for a better convergence. We also do not need to log-transform the inner-objective this time because the cross entropy loss has a log operator itself. The meta-training, i.e. training the parameters of the MLP in the MLP optimzier, is completed using a classic RMSProp optimizer with meta learning rate 0.01.

For each settings of sample sizes and noise levels, we train two MLP optimizer: one for train-by-train, and one for train-by-validation. When we test the learned MLP optimizer, we use similar settings as the inner-training problem, and we run the trajectories longer for full convergence (4000 steps for small data sets; 40000 steps for the complete data set). We run 5 independent tests and collect training accuracy and test accuracy for evaluation. The plots show the mean of the 5 tests. We have also tuned a SGD optimizer (with the same mini-batch size) by doing a grid-search of the learning rate as baseline.

D.6.2 Additional Results

Optimizing step size for quadratic objective We try experiments for the same settings of the initial η_0 and inner training length t for all of three implementations (our hand-derived GD version, Tensorflow GD version and the Tensorflow RMSProp version). We do 1000 meta-training steps for all the experiments.

For both Tensorflow versions, we always see infinite meta-objectives if η_0 is large

or t is large, whose meta-gradient is usually treated as zero, so the training get stuck and never converge. Even for the case that both η_0 and t is small, it still has very large meta-objectives (the scale of a few hundreds), and that is why we also try RMSProp, which should be more robust against the gradient scales. Our hand-derived version, however, does not have the numerical issues and can always converge to the optimal η^*. The detailed convergence is summarized in Tab D.1 and Tab D.2. Note that the optimal η^* is usually around 0.03 under our settings.

Table D.1: Whether the implementation converges for different t (fixed $\eta_0 = 0.1$)

t	10	20	40	80
Ours	✓	✓	✓	✓
Tensorflow GD	×	×	×	×
Tensorflow RMSProp	✓	✓	×	×

Table D.2: Whether the implementation converges for different η_0 (fixed $t = 40$)

η_0	0.001	0.01	0.1	1
Ours	✓	✓	✓	✓
Tensorflow GD	×	×	×	×
Tensorflow RMSProp	✓	✓	×	×

Train-by-train vs. train-by-validation, MLP optimizer on MNIST We also do additional experiments on training an MLP optimizer on the MNIST classification problem. We first try using all samples under the 20% noised setting. The results are shown in Fig D.1. The train-by-train setting can perform well if we have a large data set, but since there is also noise in the data, the train-by-train model still overfits and is slightly worse than the train-by-validation model.

We then try an intermediate sample size 12000. The results are shown in Fig D.2 (no noise) and Fig D.3 (20% noise). We can see that as the theory predicts, as the amount of data increases (from 1000 samples to 12000 samples and then to 60000

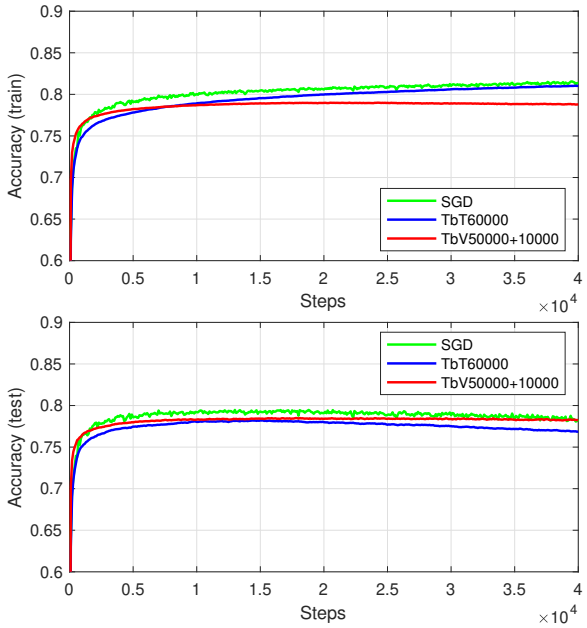


FIGURE D.1: Training and testing accuracy for different models (all samples, 20% noise)

samples) the gap between train-by-train and train-by-validation decreases. Also, when we condition on the same number of samples, having additional label noise always makes train-by-train model much worse compared to train-by-validation.

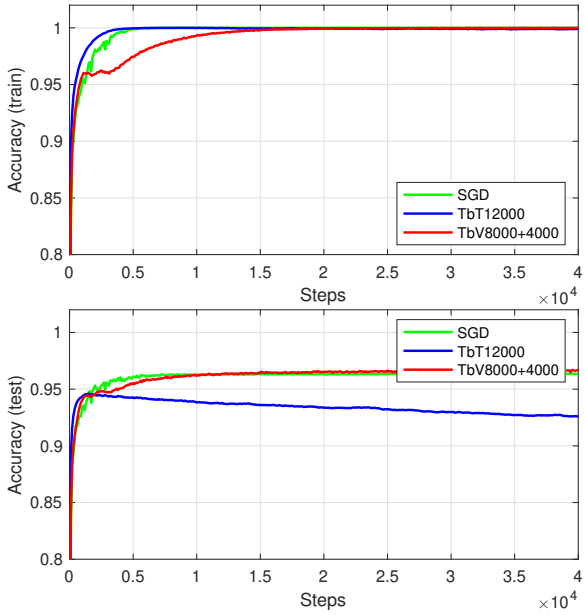


FIGURE D.2: Training and testing accuracy for different models (12000 samples, no noise)

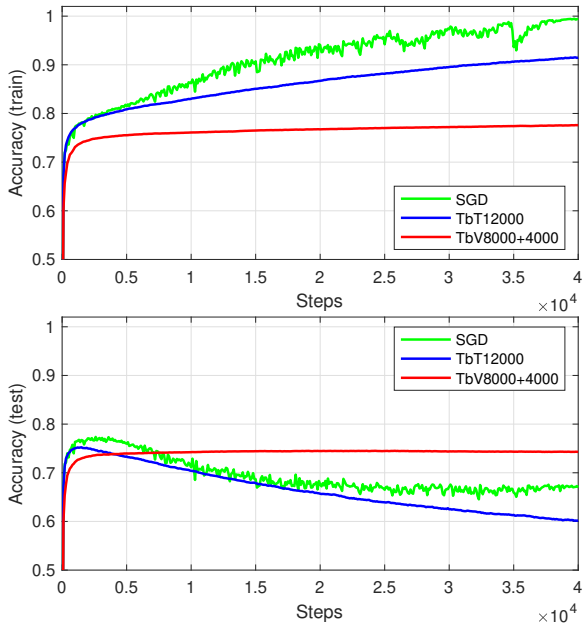


FIGURE D.3: Training and testing accuracy for different models (12000 samples, 20% noise)

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016), "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283.

Aharon, M., Elad, M., and Bruckstein, A. (2006a), "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, 54, 4311–4322.

Aharon, M., Elad, M., and Bruckstein, A. M. (2006b), "On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them," *Linear Algebra and its Applications*, 416, 48–67, Special Issue devoted to the Haifa 2005 conference on matrix theory.

Alabi, D., Kalai, A. T., Ligett, K., Musco, C., Tzamos, C., and Vitercik, E. (2019), "Learning to prune: Speeding up repeated computations," *arXiv preprint arXiv:1904.11875*.

Allen-Zhu, Z., Li, Y., and Song, Z. (2019), "A Convergence Theory for Deep Learning via Over-Parameterization," in *Proceedings of the 36th International Conference on Machine Learning*, eds. K. Chaudhuri and R. Salakhutdinov, vol. 97 of *Proceedings of Machine Learning Research*, pp. 242–252, PMLR.

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and De Freitas, N. (2016), "Learning to learn by gradient descent by gradient descent," in *Advances in neural information processing systems*, pp. 3981–3989.

Arora, S., Ge, R., and Moitra, A. (2013), "New Algorithms for Learning Incoherent and Overcomplete Dictionaries," .

Arora, S., Bhaskara, A., Ge, R., and Ma, T. (2014), "More Algorithms for Provable Dictionary Learning," *CoRR*, abs/1401.0579.

Arora, S., Ge, R., Ma, T., and Moitra, A. (2015), "Simple, Efficient, and Neural Algorithms for Sparse Coding," *CoRR*, abs/1503.00778.

Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. (2016), "A latent variable model approach to pmi-based word embeddings," *Transactions of the Association for Computational Linguistics*, 4, 385–399.

Arora, S., Liang, Y., and Ma, T. (2017), "A simple but tough-to-beat baseline for sentence embeddings," in *International conference on learning representations*.

Arora, S., Khodak, M., Saunshi, N., and Vodrahalli, K. (2018a), "A compressed sensing view of unsupervised text embeddings, bag-of-n-grams, and LSTMs," in *International Conference on Learning Representations*.

Arora, S., Li, Z., and Lyu, K. (2018b), "Theoretical analysis of auto rate-tuning by batch normalization," *arXiv preprint arXiv:1812.03981*.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. (2019), "On exact computation with an infinitely wide neural net," *Advances in neural information processing systems*, 32.

Arora, S., Li, Z., and Panigrahi, A. (2022), "Understanding Gradient Descent on the Edge of Stability in Deep Learning," in *Proceedings of the 39th International Conference on Machine Learning*, eds. K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, vol. 162 of *Proceedings of Machine Learning Research*, pp. 948–1024, PMLR.

Bach, F., Mairal, J., and Ponce, J. (2008), "Convex Sparse Matrix Factorizations," .

Bai, Y., Chen, M., Zhou, P., Zhao, T., Lee, J. D., Kakade, S., Wang, H., and Xiong, C. (2020), "How Important is the Train-Validation Split in Meta-Learning?" *arXiv preprint arXiv:2010.05843*.

Balcan, M.-F., Nagarajan, V., Vitercik, E., and White, C. (2016a), "Learning-theoretic foundations of algorithm configuration for combinatorial partitioning problems," *arXiv preprint arXiv:1611.04535*.

Balcan, M.-F., Sandholm, T., and Vitercik, E. (2016b), "Sample complexity of automated mechanism design," *arXiv preprint arXiv:1606.04145*.

Balcan, M.-F., Sandholm, T., and Vitercik, E. (2018a), "A general theory of sample complexity for multi-item profit maximization," in *Proceedings of the 2018 ACM Conference on Economics and Computation*, pp. 173–174.

Balcan, M.-F., Dick, T., Sandholm, T., and Vitercik, E. (2018b), "Learning to branch," *arXiv preprint arXiv:1803.10150*.

Bandeira, A. S., Fickus, M., Mixon, D. G., and Wong, P. (2012), "The road to deterministic matrices with the restricted isometry property," .

Bansal, Y., Kaplun, G., and Barak, B. (2020), "For self-supervised learning, rationality implies generalization, provably," *arXiv preprint arXiv:2010.08508*.

Baraniuk, R., Davenport, M., DeVore, R., and Wakin, M. (2008), "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, 28, 253–263.

Belinkov, Y., Durrani, N., Dalvi, F., Sajjad, H., and Glass, J. (2017), "What do neural machine translation models learn about morphology?" *arXiv preprint arXiv:1704.03471*.

Bello, I., Zoph, B., Vasudevan, V., and Le, Q. V. (2017), "Neural optimizer search with reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 459–468, JMLR. org.

Bengio, S., Bengio, Y., Cloutier, J., and Gecsei, J. (1992), "On the optimization of a synaptic learning rule," in *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, vol. 2.

Bengio, Y., Bengio, S., and Cloutier, J. (1990), *Learning a synaptic learning rule*, Citeseer.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021), "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020a), "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems*, eds. H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, vol. 33, pp. 1877–1901, Curran Associates, Inc.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020b), "Language models

are few-shot learners," *Advances in neural information processing systems*, 33, 1877–1901.

Cai, T. T. and Wang, L. (2011), "Orthogonal Matching Pursuit for Sparse Signal Recovery With Noise," *IEEE Transactions on Information Theory*, 57, 4680–4688.

Candes, E. and Tao, T. (2005), "Decoding by linear programming," *IEEE Transactions on Information Theory*, 51, 4203–4215.

Candes, E., Romberg, J., and Tao, T. (2006), "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, 52, 489–509.

Candes, E. J. and Tao, T. (2006), "Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies?" *IEEE Transactions on Information Theory*, 52, 5406–5425.

Candes, E. J. and Wakin, M. B. (2008), "An Introduction To Compressive Sampling," *IEEE Signal Processing Magazine*, 25, 21–30.

Cao, S., Xu, P., and Clifton, D. A. (2022), "How to understand masked autoencoders," *arXiv preprint arXiv:2202.03670*.

Carratino, L., Cissé, M., Jenatton, R., and Vert, J.-P. (2020), "On mixup regularization," *arXiv preprint arXiv:2006.06049*.

Chen, S. and Donoho, D. (1994), "Basis pursuit," in *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 41–44, IEEE.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020), "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR.

Chidambaram, M., Wu, C., Cheng, Y., and Ge, R. (2023), "Hiding Data Helps: On the Benefits of Masking for Sparse Coding," *arXiv preprint arXiv:2302.12715*.

Chizat, L. and Bach, F. (1812), "A note on lazy training in supervised differentiable programming.(2018)," *arXiv preprint arXiv:1812.07956*.

Chizat, L. and Bach, F. (2018), "On the global convergence of gradient descent for over-parameterized models using optimal transport," *Advances in neural information processing systems*, 31.

Dangel, F., Harmeling, S., and Hennig, P. (2020), "Modular Block-diagonal Curvature Approximations for Feedforward Architectures," in *International Conference on Artificial Intelligence and Statistics*, pp. 799–808.

Daubechies, I., Defrise, M., and Mol, C. D. (2003), "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, 57.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018), "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019), "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, Association for Computational Linguistics.

Ding, T., Li, D., and Sun, R. (2019), "Spurious local minima exist for almost all over-parameterized neural networks," *Optimization online*.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. (2017), "Sharp Minima Can Generalize For Deep Nets," in *International Conference on Machine Learning*, pp. 1019–1028.

Donoho, D. (2006), "Compressed sensing," *IEEE Transactions on Information Theory*, 52, 1289–1306.

Donoho, D., Elad, M., and Temlyakov, V. (2006), "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transactions on Information Theory*, 52, 6–18.

Donoho, D. L., Maleki, A., and Montanari, A. (2009), "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, 106, 18914–18919.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. (2018), "Gradient descent provably optimizes over-parameterized neural networks," *arXiv preprint arXiv:1810.02054*.

Duarte, M. F. and Eldar, Y. C. (2011), "Structured Compressed Sensing: From Theory to Applications," *CoRR*, abs/1106.6224.

Dziugaite, G. K. and Roy, D. M. (2017), "Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data," in *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI*.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least angle regression," *The Annals of Statistics*, 32, 407 – 499.

Engan, K., Aase, S., and Hakon Husoy, J. (1999), "Method of optimal directions for frame design," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, vol. 5, pp. 2443–2446 vol.5.

Fedus, W., Zoph, B., and Shazeer, N. (2021), "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *J. Mach. Learn. Res*, 23, 1–40.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. (2020), "Sharpness-aware minimization for efficiently improving generalization," *arXiv preprint arXiv:2010.01412*.

Fort, S. and Ganguli, S. (2019), "Emergent properties of the local geometry of neural loss landscapes," *arXiv preprint arXiv:1910.05929*.

Franceschi, L., Donini, M., Frasconi, P., and Pontil, M. (2017), "Forward and reverse gradient-based hyperparameter optimization," *arXiv preprint arXiv:1703.01785*.

Ge, R., Kakade, S. M., Kidambi, R., and Netrapalli, P. (2019), "The Step Decay Schedule: A Near Optimal, Geometrically Decaying Learning Rate Procedure For Least Squares," in *Advances in Neural Information Processing Systems*, pp. 14951–14962.

Geng, Q., Wang, H., and Wright, J. (2011), "On the Local Correctness of L^1 Minimization for Dictionary Learning," *CoRR*, abs/1101.5672.

George, T., Laurent, C., Bouthillier, X., Ballas, N., and Vincent, P. (2018), "Fast approximate natural gradient descent in a kronecker factored eigenbasis," in *Advances in Neural Information Processing Systems*, pp. 9550–9560.

Ghorbani, B., Krishnan, S., and Xiao, Y. (2019), "An Investigation into Neural Net Optimization via Hessian Eigenvalue Density," in *International Conference on Machine Learning*, pp. 2232–2241.

Glorot, X. and Bengio, Y. (2010), "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256.

Golmant, N., Yao, Z., Gholami, A., Mahoney, M., and Gonzalez, J. (2018), "pytorch-hessian-eigentings: efficient PyTorch Hessian eigendecomposition," .

Goodfellow, I., Bengio, Y., and Courville, A. (2016), *Deep Learning*, MIT Press.

Gribonval, R. and Schnass, K. (2010), "Dictionary Identification—Sparse Matrix-Factorization via ℓ_1 -Minimization," *IEEE Transactions on Information Theory*, 56, 3523–3539.

Gribonval, R., Jenatton, R., and Bach, F. R. (2014), "Sparse and spurious: dictionary learning with noise and outliers," *CoRR*, abs/1407.5155.

Grosse, R. and Martens, J. (2016), "A kronecker-factored approximate fisher matrix for convolution layers," in *International Conference on Machine Learning*, pp. 573–582.

Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., et al. (2020), "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*.

Guo, H., Mao, Y., and Zhang, R. (2019), "MixUp as Locally Linear Out-of-Manifold Regularization," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19, AAAI Press.

Gupta, R. and Roughgarden, T. (2017), "A PAC approach to application-specific algorithm selection," *SIAM Journal on Computing*, 46, 992–1017.

Gur-Ari, G., Roberts, D. A., and Dyer, E. (2018), "Gradient descent happens in a tiny subspace," *arXiv preprint arXiv:1812.04754*.

HaoChen, J. Z., Wei, C., Gaidon, A., and Ma, T. (2021), "Provable guarantees for self-supervised deep learning with spectral contrastive loss," *Advances in Neural Information Processing Systems*, 34, 5000–5011.

Hardt, M. and Ma, T. (2016), "Identity matters in deep learning," *arXiv preprint arXiv:1611.04231*.

Harvey, N. J., Liaw, C., Plan, Y., and Randhawa, S. (2018), "Tight analyses for non-smooth stochastic gradient descent," *arXiv preprint arXiv:1812.05217*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a), "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b), "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022), "Masked Autoencoders Are Scalable Vision Learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009.

Heskes, T. (2000), "On "natural" learning and pruning in multilayered perceptrons," *Neural Computation*, 12, 881–901.

Hewitt, J. and Manning, C. D. (2019), "A structural probe for finding syntax in word representations," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138.

Hochreiter, S., Younger, A. S., and Conwell, P. R. (2001), "Learning to learn using gradient descent," in *International Conference on Artificial Neural Networks*, pp. 87–94, Springer.

Ioffe, S. and Szegedy, C. (2015), "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *International Conference on Machine Learning*, pp. 448–456.

Jacot, A., Gabriel, F., and Hongler, C. (2018), "Neural tangent kernel: Convergence and generalization in neural networks," *Advances in neural information processing systems*, 31.

Jacot, A., Gabriel, F., and Hongler, C. (2020), "The asymptotic spectrum of the Hessian of DNN throughout training," in *8th International Conference on Learning Representations, ICLR*.

Jain, P., Nagaraj, D., and Netrapalli, P. (2019), "Making the last iterate of sgd information theoretically optimal," *arXiv preprint arXiv:1904.12443*.

Jastrzebski, S., Kenton, Z., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. J. (2019), "On the Relation Between the Sharpest Directions of DNN Loss and the SGD Step Length," in *7th International Conference on Learning Representations, ICLR*.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. (2019), "Fantastic generalization measures and where to find them," *arXiv preprint arXiv:1912.02178*.

Johnson, W. B. and Lindenstrauss, J. (1984), "Extensions of Lipschitz mappings into a Hilbert space," *Contemporary mathematics*, 26, 1.

Karakida, R., Akaho, S., and Amari, S.-i. (2019a), "The Normalization Method for Alleviating Pathological Sharpness in Wide Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 6406–6416.

Karakida, R., Akaho, S., and Amari, S.-i. (2019b), "Pathological spectra of the fisher information metric and its variants in deep neural networks," *arXiv preprint arXiv:1910.05992*.

Karakida, R., Akaho, S., and Amari, S.-i. (2019c), "Universal statistics of fisher information in deep neural networks: Mean field approach," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1032–1041, PMLR.

Kawaguchi, K. (2016), "Deep learning without poor local minima," *Advances in neural information processing systems*, 29.

Kawaguchi, K. and Kaelbling, L. (2020), "Elimination of all bad local minima in deep learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 853–863, PMLR.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2017), "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," in *5th International Conference on Learning Representations, ICLR*.

Kim, T., Choi, J., Edmiston, D., and Lee, S.-g. (2020), "Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction," *arXiv preprint arXiv:2002.00737*.

Kingma, D. P. and Ba, J. (2014), "Adam: A Method for Stochastic Optimization," .

Kleinman, D. and Athans, M. (1968), "The design of suboptimal linear time-varying systems," *IEEE Transactions on Automatic Control*, 13, 150–159.

Kohler, J., Daneshmand, H., Lucchi, A., Zhou, M., Neymeyr, K., and Hofmann, T. (2018), "Towards a theoretical understanding of batch normalization," *stat*, 1050, 27.

Krause, A. and Cevher, V. (2010), "Submodular Dictionary Selection for Sparse Representation," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, p. 567–574, Madison, WI, USA, Omnipress.

Krizhevsky, A. and Hinton, G. (2009), "Learning multiple layers of features from tiny images," Tech. Rep. 0, University of Toronto, Toronto, Ontario.

Langford, J. and Seeger, M. (2001), "Bounds for averaging classifiers," Tech. rep., Carnegie Mellon University.

Laurent, B. and Massart, P. (2000), "Adaptive estimation of a quadratic functional by model selection," *Annals of Statistics*, pp. 1302–1338.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998), "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 86, 2278–2324.

Lee, J. D., Lei, Q., Saunshi, N., and Zhuo, J. (2021), "Predicting what you already know helps: Provable self-supervised learning," *Advances in Neural Information Processing Systems*, 34, 309–323.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018), "Visualizing the loss landscape of neural nets," in *Advances in Neural Information Processing Systems*, pp. 6389–6399.

Li, K. and Malik, J. (2016), "Learning to optimize," *arXiv preprint arXiv:1606.01885*.

Li, K. and Malik, J. (2017), "Learning to optimize neural nets," *arXiv preprint arXiv:1703.00441*.

Li, X., Gu, Q., Zhou, Y., Chen, T., and Banerjee, A. (2020), "Hessian based analysis of sgd for deep nets: Dynamics and generalization," in *Proceedings of the 2020 SIAM International Conference on Data Mining*, pp. 190–198, SIAM.

Liang, S., Sun, R., Lee, J. D., and Srikant, R. (2018), "Adding one neuron can eliminate all bad local minima," *Advances in Neural Information Processing Systems*, 31.

Liao, Z. and Mahoney, M. W. (2021), "Hessian eigenspectra of more realistic nonlinear models," *Advances in Neural Information Processing Systems*, 34.

Liu, D. C. and Nocedal, J. (1989), "On the limited memory BFGS method for large scale optimization," *Mathematical programming*, 45, 503–528.

Lv, K., Jiang, S., and Li, J. (2017), "Learning gradient descent: Better generalization and longer horizons," in *International Conference on Machine Learning*, pp. 2247–2255, PMLR.

Maclaurin, D., Duvenaud, D., and Adams, R. (2015), "Gradient-based hyperparameter optimization through reversible learning," in *International Conference on Machine Learning*, pp. 2113–2122.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010), "Online Learning for Matrix Factorization and Sparse Coding," *J. Mach. Learn. Res.*, 11, 19–60.

Maleki, A. and Donoho, D. L. (2010), "Optimally Tuned Iterative Reconstruction Algorithms for Compressed Sensing," *IEEE Journal of Selected Topics in Signal Processing*, 4, 330–341.

Mallat, S. and Zhang, Z. (1993), "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, 41, 3397–3415.

Martens, J. and Grosse, R. (2015), "Optimizing neural networks with kronecker-factored approximate curvature," in *International conference on machine learning*, pp. 2408–2417.

McAllester, D. A. (1999), "Some pac-bayesian theorems," *Machine Learning*, 37, 355–363.

Mei, S., Montanari, A., and Nguyen, P.-M. (2018), "A mean field view of the landscape of two-layer neural networks," *Proceedings of the National Academy of Sciences*, 115, E7665–E7671.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. (2016), "Pointer Sentinel Mixture Models," .

Metz, L., Maheswaranathan, N., Nixon, J., Freeman, D., and Sohl-Dickstein, J. (2019), "Understanding and correcting pathologies in the training of learned optimizers," in *International Conference on Machine Learning*, pp. 4556–4565.

Miaschi, A. and Dell'Orletta, F. (2020), "Contextual and Non-Contextual Word Embeddings: an in-depth Linguistic Investigation," in *Proceedings of the 5th Workshop on Representation Learning for NLP*, pp. 110–119, Online, Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013), "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015), "Human-level control through deep reinforcement learning," *nature*, 518, 529–533.

Morgenstern, J. and Roughgarden, T. (2016), "Learning simple auctions," in *Conference on Learning Theory*, pp. 1298–1318, PMLR.

Morgenstern, J. H. and Roughgarden, T. (2015), "On the pseudo-dimension of nearly optimal auctions Advances in Neural Information Processing Systems. 136–144," *Google Scholar Google Scholar Digital Library Digital Library*.

Musa, O., Jung, P., and Goertz, N. (2018), "Generalized Approximate Message Passing for Unlimited Sampling of Sparse Signals," .

Natarajan, B. K. (1995), "Sparse Approximate Solutions to Linear Systems," *SIAM Journal on Computing*, 24, 227–234.

Nikunj, S., Malladi, S., and Arora, S. (2021), "A Mathematical Exploration of Why Language Models Help Solve Downstream Tasks," in *International Conference on Learning Representations*.

Olshausen, B. A. and Field, D. J. (1997), "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision Research*, 37, 3311–3325.

Olshausen, B. A. and Field, D. J. (2004), "Sparse coding of sensory inputs," *Current opinion in neurobiology*, 14, 481–487.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018), "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*.

Pan, J., Zhou, P., and Yan, S. (2022), "Towards Understanding Why Mask-Reconstruction Pretraining Helps in Downstream Tasks," *arXiv preprint arXiv:2206.03826*.

Papyan, V. (2018), "The Full Spectrum of Deepnet Hessians at Scale: Dynamics with SGD Training and Sample Size," *arXiv preprint arXiv:1811.07062*.

Papyan, V. (2019), "Measurements of Three-Level Hierarchical Structure in the Outliers in the Spectrum of Deepnet Hessians," in *International Conference on Machine Learning*, pp. 5012–5021.

Papyan, V. (2020), "Traces of Class/Cross-Class Structure Pervade Deep Learning Spectra," *arXiv preprint arXiv:2008.11865*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017), "Automatic Differentiation in PyTorch," in *NIPS 2017 Workshop on Autodiff*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019a), "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *CoRR*, abs/1912.01703.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019b), "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc.

Pennington, J., Socher, R., and Manning, C. D. (2014), "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018), "Deep Contextualized Word Representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, Association for Computational Linguistics.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019), "Language models are unsupervised multitask learners," *OpenAI blog*, 1, 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020a), "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, 21, 1–67.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020b), "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, 21, 5485–5551.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022), "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*.

Rubinstein, R., Zibulevsky, M., and Elad, M. (2008), "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," Tech. rep., Citeseer.

Rudelson, M. and Vershynin, R. (2009), "Smallest singular value of a random rectangular matrix," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 62, 1707–1739.

Safran, I. and Shamir, O. (2018), "Spurious local minima are common in two-layer relu neural networks," in *International conference on machine learning*, pp. 4433–4441, PMLR.

Sagun, L., Bottou, L., and LeCun, Y. (2016), "Eigenvalues of the hessian in deep learning: Singularity and beyond," *arXiv preprint arXiv:1611.07476*.

Sagun, L., Evci, U., Güney, V. U., Dauphin, Y. N., and Bottou, L. (2018), "Empirical Analysis of the Hessian of Over-Parametrized Neural Networks," in *6th International Conference on Learning Representations, ICLR 2018, Workshop Track Proceedings*.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018), "How does batch normalization help optimization?" *Advances in neural information processing systems*, 31.

Saunshi, N., Plevrakis, O., Arora, S., Khodak, M., and Khandeparkar, H. (2019), "A theoretical analysis of contrastive unsupervised representation learning," in *International Conference on Machine Learning*, pp. 5628–5637, PMLR.

Saunshi, N., Malladi, S., and Arora, S. (2020), "A mathematical exploration of why language models help solve downstream tasks," *arXiv preprint arXiv:2010.03648*.

Schoenholz, S. S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. (2017), "Deep information propagation," in *International Conference on Learning Representations, ICLR*.

Shamir, O. and Zhang, T. (2013), "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes," in *International conference on machine learning*, pp. 71–79.

Simonyan, K. and Zisserman, A. (2015), "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *3rd International Conference on Learning Representations, ICLR*, eds. Y. Bengio and Y. LeCun.

Singh, S. P., Bachmann, G., and Hofmann, T. (2021), "Analytic Insights into Structure and Rank of Neural Network Hessian Maps," *Advances in Neural Information Processing Systems*, 34.

Singla, S., Wallace, E., Feng, S., and Feizi, S. (2019), "Understanding Impacts of High-Order Loss Approximations and Features in Deep Learning Interpretation," in *International Conference on Machine Learning*, pp. 5848–5856.

Skorski, M. (2019), "Chain Rules for Hessian and Higher Derivatives Made Easy by Tensor Calculus," *arXiv preprint arXiv:1911.13292*.

Soudry, D. and Hoffer, E. (2017), "Exponentially vanishing sub-optimal local minima in multilayer neural networks," *arXiv preprint arXiv:1702.05777*.

Sra, S., Nowozin, S., and Wright, S. J. (2012), *Optimization for machine learning*, Mit Press.

Sulam, J., You, C., and Zhu, Z. (2020), "Recovery and Generalization in Over-Realized Dictionary Learning," *CoRR*, abs/2006.06179.

Sulam, J., You, C., and Zhu, Z. (2022), "Recovery and generalization in over-realized dictionary learning," *Journal of Machine Learning Research*, 23, 1–23.

Tanaka, H., Shinnou, H., Cao, R., Bai, J., and Ma, W. (2020), "Document classification by word embeddings of bert," in *Computational Linguistics: 16th International Conference of the Pacific Association for Computational Linguistics, PACLING 2019, Hanoi, Vietnam, October 11–13, 2019, Revised Selected Papers 16*, pp. 145–154, Springer.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Van Durme, B., Bowman, S. R., Das, D., et al. (2019), "What do you learn from context? probing for sentence structure in contextualized word representations," *arXiv preprint arXiv:1905.06316*.

Tian, Y., Yu, L., Chen, X., and Ganguli, S. (2020), "Understanding self-supervised learning with dual deep networks," *arXiv preprint arXiv:2010.00578*.

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, 58, 267–288.

Tibshirani, R. and Wang, P. (2008), "Spatial smoothing and hot spot detection for CGH data using the fused lasso," *Biostatistics*, 9, 18–29.

Torralba, A., Fergus, R., and Freeman, W. T. (2008), "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, 30, 1958–1970.

Tosh, C., Krishnamurthy, A., and Hsu, D. (2021a), "Contrastive estimation reveals topic posterior information to linear models," *The Journal of Machine Learning Research*, 22, 12883–12913.

Tosh, C., Krishnamurthy, A., and Hsu, D. (2021b), "Contrastive learning, multi-view redundancy, and linear models," in *Algorithmic Learning Theory*, pp. 1179–1206, PMLR.

Tosh, C., Krishnamurthy, A., and Hsu, D. (2021c), "Contrastive learning, multi-view redundancy, and linear models," in *Algorithmic Learning Theory*, pp. 1179–1206, PMLR.

Tropp, J. (2006), "Just relax: convex programming methods for identifying sparse signals in noise," *IEEE Transactions on Information Theory*, 52, 1030–1051.

Tropp, J. A. and Gilbert, A. C. (2007), "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit," *IEEE Transactions on Information Theory*, 53, 4655–4666.

Tsai, Y.-H. H., Wu, Y., Salakhutdinov, R., and Morency, L.-P. (2020), "Self-supervised learning from a multi-view perspective," *arXiv preprint arXiv:2006.05576*.

Van der Maaten, L. and Hinton, G. (2008), "Visualizing data using t-SNE." *Journal of machine learning research*, 9.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017), "Attention is all you need," *Advances in neural information processing systems*, 30.

Vershynin, R. (2010), "Introduction to the non-asymptotic analysis of random matrices," *arXiv preprint arXiv:1011.3027*.

Vershynin, R. (2018a), *High-dimensional probability: An introduction with applications in data science*, vol. 47, Cambridge university press.

Vershynin, R. (2018b), *High-dimensional probability: An introduction with applications in data science*, vol. 47, Cambridge university press.

Wang, T. and Isola, P. (2020), "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *International Conference on Machine Learning*, pp. 9929–9939, PMLR.

Wang, X., Yuan, S., Wu, C., and Ge, R. (2021), "Guarantees for tuning the step size using a learning-to-learn approach," in *International Conference on Machine Learning*, pp. 10981–10990, PMLR.

Wei, C., Xie, S. M., and Ma, T. (2021), "Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning," *Advances in Neural Information Processing Systems*, 34, 16158–16170.

Wichrowska, O., Maheswaranathan, N., Hoffman, M. W., Colmenarejo, S. G., Denil, M., de Freitas, N., and Sohl-Dickstein, J. (2017), "Learned optimizers that scale and generalize," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3751–3760, JMLR. org.

Wiedemann, G., Remus, S., Chawla, A., and Biemann, C. (2019), "Does BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings," *arXiv preprint arXiv:1909.10430*.

Wu, Y., Zhu, X., Wu, C., Wang, A., and Ge, R. (2020), "Dissecting hessian: Understanding common structure of hessian in neural networks," *arXiv preprint arXiv:2010.04261*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019), "XLNet: Generalized Autoregressive Pretraining for Language Understanding," in *Advances in Neural Information Processing Systems*, eds. H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, vol. 32, Curran Associates, Inc.

Yao, Z., Gholami, A., Lei, Q., Keutzer, K., and Mahoney, M. W. (2018), "Hessian-based analysis of large batch training and robustness to adversaries," in *Advances in Neural Information Processing Systems*, pp. 4949–4959.

Yao, Z., Gholami, A., Keutzer, K., and Mahoney, M. (2019), "PyHessian: Neural networks through the lens of the Hessian," *arXiv preprint arXiv:1912.07145*.

Yin, W., Osher, S., Goldfarb, D., and Darbon, J. (2008), "Bregman Iterative Algorithms for ℓ_1-Minimization with Applications to Compressed Sensing," *SIAM Journal on Imaging Sciences*, 1, 143–168.

Zhang, L., Deng, Z., Kawaguchi, K., Ghorbani, A., and Zou, J. (2020), "How does mixup help with robustness and generalization?" *arXiv preprint arXiv:2010.04819*.

Zhang, L., Deng, Z., Kawaguchi, K., and Zou, J. (2022a), "When and how mixup improves calibration," in *International Conference on Machine Learning*, pp. 26135–26160, PMLR.

Zhang, R., Shen, J., Wei, F., Li, X., and Sangaiah, A. K. (2017), "Medical image classification based on multi-scale non-negative sparse coding," *Artificial intelligence in medicine*, 83, 44–51.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. (2022b), "Opt: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068*.

Zhang, T. and Hashimoto, T. (2021), "On the inductive bias of masked language modeling: From statistical to syntactic dependencies," *arXiv preprint arXiv:2104.05694*.

Zhou, M., Chen, H., Ren, L., Sapiro, G., Carin, L., and Paisley, J. (2009), "Non-Parametric Bayesian Dictionary Learning for Sparse Image Representations," in *Advances in Neural Information Processing Systems*, eds. Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, vol. 22, Curran Associates, Inc.

Zhu, S. (2012), "A short note on the tail bound of wishart distribution," *arXiv preprint arXiv:1212.5860*.

Zhu, X., Wang, Z., Wang, X., Zhou, M., and Ge, R. (2022), "Understanding Edge-of-Stability Training Dynamics with a Minimalist Example," *arXiv preprint arXiv:2210.03294*.

Biography

Chenwei Wu is a Ph.D. candidate in Computer Science at Duke University. He is advised by Rong Ge. He received his bachelor's degree from Institute for Interdisciplinary Information Sciences (Yao Class), Tsinghua University in 2018. He received Best Reviewers award for ICML 2019, was among top 10% high-scoring reviewers for NeurIPS 2020, and received Outstanding Reviewers award for ICLR 2021.