Random Masking Finds Winning Tickets for Parameter Efficient Fine-tuning

Jing Xu [1] **Jingzhao Zhang** [1 2 3]

Abstract

Fine-tuning large language models (LLM) can be costly. Parameter-efficient fine-tuning (PEFT) addresses the problems by training a fraction of the parameters, whose success reveals the expressiveness and flexibility of pretrained models. This paper studies the limit of PEFT, by further simplifying its design and reducing the number of trainable parameters beyond standard setups. To this end, we use Random Masking to fine-tune the pretrained model. Despite its simplicity, we show that Random Masking is surprisingly effective: with a larger-than-expected learning rate, Random Masking can match the performance of standard PEFT algorithms such as LoRA on various tasks, using fewer trainable parameters. We provide both empirical and theoretical explorations into the success of Random Masking. We show that masking induces a flatter loss landscape and more distant solutions, which allows for and necessitates large learning rates.

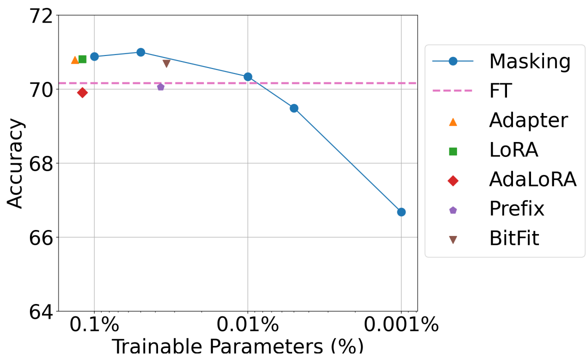


Figure 1: **The average performance of PEFT methods over with various numbers of trainable parameters.** Masking stands for our Random Masking method; FT stands for full parameter fine-tuning; Prefix stands for Prefix-Tuning. The metrics are calculated on 11 datasets using OPT-1.3b. Despite its simple design, Random Masking achieves competitive performance with fewer trainable parameters.

1. Introduction

Large-scale pretrained models (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023a) have revolutionized deep learning, demonstrating remarkable capabilities in various domains such as natural language processing and computer vision. These models use an extensive number of parameters to capture complex patterns in data. Despite their success, the intensive resources required for utilizing these models pose significant challenges, especially in the setting where they have to be *fine-tuned* to adapt to downstream data or align with human behaviors. To reduce the computational and memory demands, researchers have developed various *parameter efficient fine-tuning (PEFT)* algorithms, such as LoRA (Hu et al., 2021), adapter (Houlsby et al., 2019), prompt tuning (Li & Liang, 2021; Lester et al., 2021). These methods have seen widespread application for both language (Shi et al., 2023; Lialin et al., 2023; Liu et al., 2022) and vision tasks (Sung et al., 2022; Lin et al., 2023).

The success of PEFT using a remarkably small fraction of parameters inspired research efforts to understand the phenomenon. For example, Aghajanyan et al. (2020) and Malladi et al. (2023b) show that though the pretrained model parameters live in a high dimensional space, fine-tuning tasks have low complexity in terms of intrinsic dimensions. Additionally, research indicates that pretrained networks have better optimization landscapes compared with random initialized networks (Hao et al., 2019; Zhou & Srikumar, 2021), making them easier to fit the downstream datasets. Furthermore, Su et al. (2023b) highlights the importance of model scaling in PEFT, showing that it can even mitigate the impact of design differences among PEFT methods.

Motivated by the observations and analyses on the effectiveness of PEFT, our work hopes to take a step forward and explore the performance limit of PEFT. More specif-

[1] Institute for Interdisciplinary Information Sciences, Tsinghua University, China [2] Shanghai Qizhi Institute [3] Shanghai AI Laboratory. Correspondence to: Jing Xu <xujing21@mails.tsinghua.edu.cn>, Jingzhao Zhang <jingzhaoz@mail.tsinghua.edu.cn>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

ically, *is there any room to further reduce the parameters and simplify the design of PEFT modules?*

Inspired by the success of neural network pruning and lottery ticket hypothesis, this paper studies a PEFT method, which we call *Random Masking*. Specifically, Random Masking involves applying a random binary mask on the model parameters, and only training the unmasked parameters during fine-tuning. Random Masking provides a convenient way for us to reduce the trainable parameters beyond the current limit, and moreover, it has a simple design that incorporates nearly no inductive bias about the model architecture or the task.

Random Masking is typically treated as a baseline with subpar performance in previous research. However, our experiments reveal a surprising phenomenon that in fine-tuning LLM to SuperGLUE datasets, Random Masking can match the performance of full-parameter fine-tuning and standard PEFT methods across various model scales. The key to the success of Random Masking is the selection of an appropriate learning rate. Specifically, we find that sparser masking requires aggressive learning rates. The optimal learning rate can be up to $1\mathrm{e}{-1}$, a value that typically results in divergence for standard PEFT methods.

The effectiveness of Random Masking suggests a greater expressive capacity of pretrained models than previously recognized. Remarkably, our experiments show that with as little as 0.001% of the parameters being trainable, Random Masking can still achieve a non-trivial accuracy. This ratio of trainable parameters is about 100 times smaller than that in LoRA. These results imply a large parameter redundancy in practical PEFT methods.

We provide a thorough investigation into the success of Random Masking. Empirically, we demonstrate that masking induces a flatter loss landscape, in terms of the loss Hessian spectrum. This explains why aggressive learning rates do not result in divergence. Simultaneously, we illustrate that a flatter loss landscape gives rise to more distant solutions, which explains why large learning rates are necessary for sparse masking.

Theoretically, we analyze the overparameterized linear regression model. We prove a bound on the Hessian eigenvalues of masked models using matrix concentration bounds, revealing a decay in eigenvalues as the masking becomes sparser. We also prove that smaller Hessian eigenvalues allow for larger learning rates and induce more distant solutions. These findings align with our empirical results and provide a cohesive explanation of how Random Masking influences learning dynamics.

Our analysis reveals a trade-off between the expressiveness of pretrained models and the difficulty of optimization. Randomly Masked model has a benign loss landscape by sacrificing model expressivity. The remaining model capacity is insufficient for difficult tasks such as pretraining; however, it is already enough to fit a pretrained LLM on various fine-tuning tasks. This also sheds light on why PEFT methods outperform full-parameter fine-tuning in the low data regime (Zaken et al., 2021), since they trade-off the redundant parameters for a better optimization landscape.

We summarize our contributions as follows:

- We show that Random Masking with a properly tuned learning rate can achieve comparable performance to standard PEFT methods on SuperGLUE benchmarks, with a significantly reduced trainable parameter count.
- We provide extensive experiment results to show that the benign loss landscape and expressive power of pretrained models are the key factors in the success of Random Masking.
- We provide theoretical studies on the overparameterized linear regression model, elucidating the interplay between learning rate tuning and Random Masking.

2. Related Works

Parameter Efficient Fine-tuning. PEFT has garnered significant attention, leading to a diverse array of algorithmic and architectural innovations. The first wave of PEFT methods involves integrating small trainable adapters (Houlsby et al., 2019; Pfeiffer et al., 2020; Rücklé et al., 2020; Karimi Mahabadi et al., 2021; He et al., 2021b; Zhu et al., 2021; Jie & Deng, 2022; Zhang et al., 2023e; Gao et al., 2023) into the pretrained networks. Another line of methods adds trainable continuous modules to the prompt, which is called prompt tuning or prefix tuning (Li & Liang, 2021; Lester et al., 2021; Jia et al., 2022; Liu et al., 2023). The seminal work Hu et al. (2021) proposes the LoRA algorithm, which has become one of the most widely used PEFT methods due to its performance and versatility. A series of works propose variants of the LoRA algorithm, aiming to further reduce the trainable parameter count (Zhang et al., 2023b; Kopiczko et al., 2023; Ding et al., 2023), enhance the expressiveness of low rank structures (Koohpayegani et al., 2023; Zi et al., 2023), implement adaptive parameter allocation (Zhang et al., 2023a;d), and combine LoRA with other techniques such as quantization (Dettmers et al., 2023; Xu et al., 2023) and pruning (Zhang et al., 2023c).

Besides directly designing PEFT modules, several studies build unified frameworks for PEFT modules (He et al., 2021a; Mao et al., 2021; Ding et al., 2022; Chen et al., 2023), thereby facilitating more efficient configuration selection (Zhou et al., 2023; Hu et al., 2022). Another line of works focuses on designing lightweight optimization algorithms tailored for tuning large models (Malladi et al.,

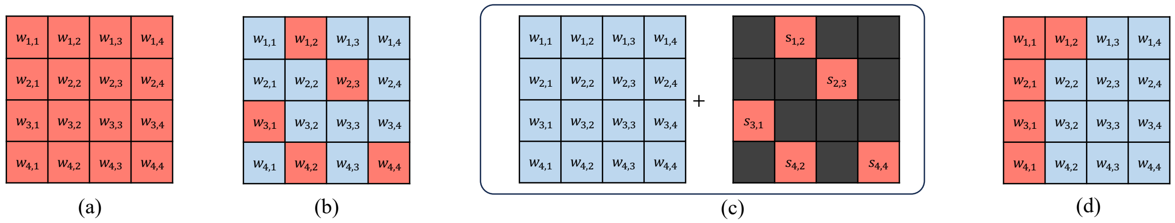

$w_{1,1}$ $w_{1,2}$ $w_{1,3}$ $w_{1,4}$
$w_{2,1}$ $w_{2,2}$ $w_{2,3}$ $w_{2,4}$
$w_{3,1}$ $w_{3,2}$ $w_{3,3}$ $w_{3,4}$
$w_{4,1}$ $w_{4,2}$ $w_{4,3}$ $w_{4,4}$
(a)
$w_{1,1}$ $w_{1,2}$ $w_{1,3}$ $w_{1,4}$
$w_{2,1}$ $w_{2,2}$ $w_{2,3}$ $w_{2,4}$
$w_{3,1}$ $w_{3,2}$ $w_{3,3}$ $w_{3,4}$
$w_{4,1}$ $w_{4,2}$ $w_{4,3}$ $w_{4,4}$
(b)
$w_{1,1}$ $w_{1,2}$ $w_{1,3}$ $w_{1,4}$
$w_{2,1}$ $w_{2,2}$ $w_{2,3}$ $w_{2,4}$
$w_{3,1}$ $w_{3,2}$ $w_{3,3}$ $w_{3,4}$
$w_{4,1}$ $w_{4,2}$ $w_{4,3}$ $w_{4,4}$
+
$s_{1,2}$
$s_{2,3}$
$s_{3,1}$
$s_{4,2}$ $s_{4,4}$
(c)
$w_{1,1}$ $w_{1,2}$ $w_{1,3}$ $w_{1,4}$
$w_{2,1}$ $w_{2,2}$ $w_{2,3}$ $w_{2,4}$
$w_{3,1}$ $w_{3,2}$ $w_{3,3}$ $w_{3,4}$
$w_{4,1}$ $w_{4,2}$ $w_{4,3}$ $w_{4,4}$
(d)

Figure 2: **Illustration of the masking methods.** The red grids indicate trainable parameters and the blue grids indicate frozen parameters. (a) Full parameter fine-tuning of $\boldsymbol{W}$. (b) The Random Masking of W, which is the main PEFT algorithm in this paper. (c) Implementation of Random masking of $\boldsymbol{W}$ via a sparse matrix $\boldsymbol{S}$ that is stored compactly as vectors. (d) The Structured Masking of $\boldsymbol{W}$, for ablation studies in Section 4.4.

2023a; Zelikman et al., 2023; Lv et al., 2023).

Masking and Pruning Methods. Masking (Sung et al., 2021; Jaiswal et al., 2022; Xu et al., 2021; Nikdan et al., 2024) is a key component in various PEFT methods, and is inherently related to neural network pruning and lottery ticket hypothesis (Han et al., 2015; Molchanov et al., 2017; Liu et al., 2017; Frankle & Carbin, 2018). Zaken et al. (2021) proposes BitFit, which implicitly masks out the model weights except for the bias vector. Some works propose algorithms to train a masking matrix (Guo et al., 2020; Zhao et al., 2020; Li et al., 2022) for fine-tuning the networks. Compared with these approaches, Random Masking in our paper does not require assigning or training the mask, and incorporates minimal inductive bias into algorithm design.

Masking is leveraged in (Su et al., 2023a) to enable a small trainable parameter count, but they focus on adding masking to the PEFT modules rather than the pretrained networks. Aghajanyan et al. (2020) apply a random projection method to calculate the intrinsic dimension of pretrained LLMs, which shares conceptual similarities with Random Masking in our paper. However, they define the intrinsic dimension as the parameter number that achieves 90% of the accuracy of full fine-tuning, while we show that Random Masking can achieve the same level of accuracy as full fine-tuning.

3. Random Masking and its Implementation

Let $\mathcal{N}$ denote a pretrained neural network and $\mathcal{W} = \{\boldsymbol{W}_1, \cdots, \boldsymbol{W}_k\}$ denote the parameters in $\mathcal{N}$. Given a dataset $\mathcal{D}$ and loss function $\ell(\mathcal{D}, \mathcal{W})$, fine-tuning $\mathcal{N}$ on $\mathcal{D}$ can be formulated as

$$\min_{\{\Delta_i\}} \ell(\mathcal{D}, \{\boldsymbol{W}_1 + \Delta_1, \cdots, \boldsymbol{W}_k + \Delta_k\}),$$

where Δ_i denotes the weight increment of each module, sharing the same dimensions as $\boldsymbol{W}_i$. Δ_i are zero-initialized to make sure that fine-tuning starts from the pretrained weights $\mathcal{W}$.

Conceptually, Random masking applies a random mask $\boldsymbol{M}_i$ to the requires_grad field of each parameter $\boldsymbol{W}_i$ in the pretrained models (See Figure 2(b)). This operation freezes the masked elements of the weight tensors, allowing only the unmasked elements to be optimized in the fine-tuning process. The elements of $\boldsymbol{M}_i$ are sampled i.i.d. from Ber(p), i.e., Bernoulli distribution of parameter p, where $p \in [0, 1]$ denotes the probability that a certain parameter is not masked. The mask matrices $\boldsymbol{M}_i$ are generated at initialization and fixed throughout fine-tuning.

Directly storing the mask matrix $\boldsymbol{M}_i$ causes large storage and computational burden. Therefore, we implement the sparse parameter update with a sparse matrix $\boldsymbol{S}_i$. This matrix consists of the coordinates of unmasked positions, that are determined by $\boldsymbol{M}_i$, and the tunable weights, both of which are stored compactly as vectors. Therefore, Random Masking can be formulated as

$$\min_{\{\boldsymbol{S}_i\}} \ell(\mathcal{D}, \{\boldsymbol{W}_1 + \boldsymbol{S}_1, \cdots, \boldsymbol{W}_k + \boldsymbol{S}_k\}).$$

One can apply off-the-shelf sparse matrix cuda libraries (Gale et al., 2020; Nikdan et al., 2024) to implement the sparse matrix $\boldsymbol{S}_i$ and solve the optimization problem.

Random masking serves as an idealized baseline to study the parameter number in PEFT, for the following two reasons. Firstly, Random Masking is flexible, since the number of trainable parameters can be manipulated by adjusting the value of p. Secondly, Random Masking is one of the most straightforward PEFT methods, introducing minimal inductive bias about the pretrained networks. This can eliminate the confounding factors, such as architecture and algorithm design, in analyzing the effect of parameter numbers.

4. Experiments

This section presents the empirical findings of Random Masking. We first outline the experiment setups and present the main results. Then we provide in-depth analyses to explore the underlying mechanisms of Random Masking.

Table 1: **Random Masking achieves comparable test accuracy with fewer trainable parameters.** This table displays the test performance of different methods. Here, FT stands for full parameter fine-tuning, Masking stands for Random Masking. Params stands for the trainable parameter ratio, which is the number of trainable parameters divided by the total parameter count of the original pretrained models. The complete results are provided in Table 5.

Model	Method	Params	SST-2	RTE	WSC	WiC	CB	BoolQ	MultiRC	COPA	ReCoRD	SQuAD	DROP	Avg
OPT-125m	FT	100%	88.1	63.5	63.5	60.3	81.0	62.9	64.7	66.0	50.8	62.4	22.8	62.36
	LoRA	0.235%	86.5	59.9	63.5	59.6	82.1	63.6	64.2	67.3	51.2	62.9	21.6	62.04
	Masking	0.1%	87.3	60.8	62.2	60.2	82.7	63.6	63.1	67.3	51.3	61.6	22.6	62.06
	Masking	0.01%	86.1	59.1	63.5	60.3	73.8	62.9	63.4	68.3	51.4	55.7	22.1	60.59
	Masking	0.001%	84.7	56.1	60.3	55.8	70.8	61.6	59.5	69.3	51.2	41.9	16.1	57.03
OPT-1.3b	FT	100%	93.7	70.5	63.1	62.7	85.7	69.5	67.6	76.7	71.8	81.2	29.3	70.16
	LoRA	0.120%	93.4	72.6	63.5	65.5	78.6	71.4	69.9	81.0	71.2	82.1	29.9	70.81
	Masking	0.1%	93.3	72.7	63.8	62.3	89.9	71.5	68.3	75.3	71.7	81.1	29.7	70.88
	Masking	0.01%	92.6	70.0	63.5	62.7	82.1	71.5	68.8	77.7	71.5	81.4	31.9	70.34
	Masking	0.001%	92.7	65.0	63.5	60.4	74.4	67.1	59.0	74.3	71.0	77.6	28.5	66.68
OPT-13b	FT	100%	94.9	81.1	62.5	65.4	81.0	79.8	76.1	89.3	81.3	87.3	35.3	75.82
	LoRA	0.051%	95.0	83.8	63.5	65.2	79.8	81.3	73.2	88.0	81.4	88.6	34.7	75.86
	Masking	0.1%	95.1	80.6	59.6	65.5	84.5	79.6	75.8	89.3	81.6	88.1	34.4	75.83
	Masking	0.01%	94.8	82.7	59.9	66.0	88.7	79.7	73.4	87.0	81.6	87.6	35.3	76.06
	Masking	0.001%	95.1	80.1	60.6	65.4	85.7	78.7	73.2	87.7	81.6	86.0	32.6	75.15

Finally, we perform various ablation studies to validate the robustness of Random Masking. Code is available at https://github.com/JingXuTHU/Random-Masking-Finds-Winning-Tickets-for-Parameter-Efficient-Fine-tuning.

4.1. Setups

Models and Datasets. We choose the OPT model family (Zhang et al., 2022) as the pretrained LLMs, using three different model scales: 125m, 1.3b and 13b. We conduct the experiments on a diverse range of datasets and tasks, including 8 datasets in the SuperGLUE benchmark (Wang et al., 2019) and three additional datasets. In line with the approach in Malladi et al. (2023a), we randomly sample 1000 data points from each dataset's original training split for training, 500 data points for validation, and randomly sample 1000 data points from its original validation split for testing. F1 score is used as the metric for SQuAD and DROP, and test accuracy is used for other datasets. We also use the same prompt templates as in Malladi et al. (2023a).

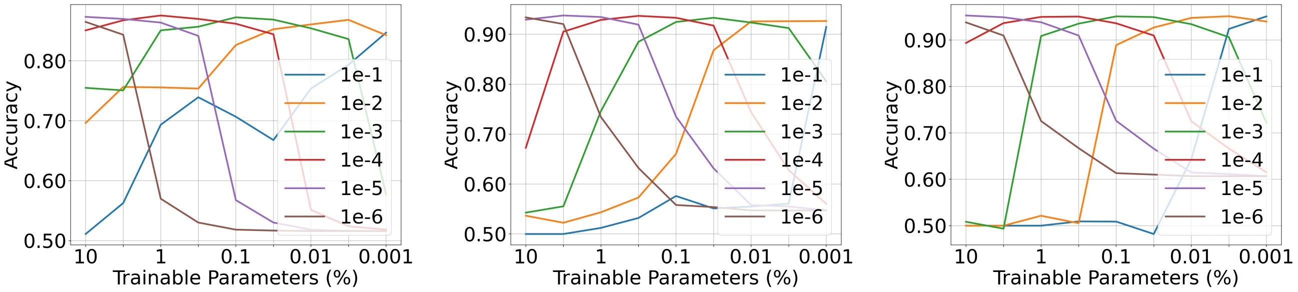
Methods. We conduct experiments using Random Masking, and consider various baselines including full parameter fine-tuning, LoRA (Hu et al., 2021). We also experiment with other baselines including Adapter (Houlsby et al., 2019), Prefix-Tuning (Li & Liang, 2021), BitFit (Zaken et al., 2021) and AdaLoRA (Zhang et al., 2023d), the results of which are given in Appendix B.1. For Random Masking, we choose the trainable parameter ratio for from $\{10\%, 5\%, 1\%, 0.5\%, 0.1\%, 0.05\%, 0.01\%, 0.005\%, 0.001\%\}$, and implement the sparse matrix operation using the spops library (Nikdan et al., 2024). We choose $r = 8$ and $\alpha = 16$ for LoRA. Following the original implementations of Lora (Hu et al., 2021), we apply LoRA and Random Masking only to the query and value matrix in each attention layer.

4.2. Main Results

Our experiments raise the following two major observations on Random Masking.

Random Masking achieves on-par performance with the baselines. In Table 1, we report the test performance of different methods, obtained using optimal grid-searched learning rates. Complete results for Random Masking with different trainable parameter ratios are provided in Table 5 in Appendix B. The results indicate that despite its simple design, Random Masking achieves comparable performance with baselines across different model scales, using a significantly smaller trainable parameter ratio. Additionally, we note that larger models are more amenable to sparser masking. Take Random Masking on OPT-13b model with trainable parameter ratios of 0.1% and 0.001% as an example: despite having a hundredfold difference in trainable parameter count, the latter one exhibits performances that are within a 2% margin of the former.

Sparse Random Masking necessitates significantly larger learning rates. We plot how the performance of Random Masking varies with different learning rates in Figure 3.

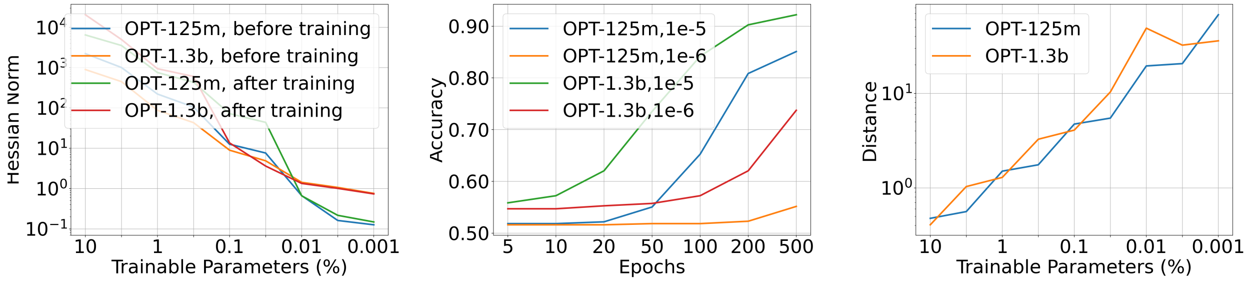


SST-2, OPT-125m

SST-2, OPT-1.3b

SST-2, OPT-13b

Figure 3: **The accuracy of Random Masking on SST-2 dataset with different learning rates.** The figure shows that the accuracy remains steady despite the small number of trainable parameters, as long as using an appropriate learning rate. As the trainable parameter ratio becomes smaller, the optimal learning rate becomes larger. The complete results of SuperGLUE benchmark are given in Figure 6, 7 and 8.



(a)

(b)

(c)

Figure 4: **Investigations into the training mechanism behind Random Masking. (a).** Smaller trainable parameter ratio induces smaller hessian ℓ_2 norm. **(b).** Longer training steps compensate small learning rates. **(c).** Smaller trainable parameter ratio gives more distant solutions. These figures present the results on SST-2 datasets. Additional Results on other datasets can be found in Figure 9, 10 and 11.

The optimal learning rates for different methods are listed in Table 6 and 7. These results highlight the critical role of an appropriate learning rate for the success of Random Masking. Our findings indicate that Random Masking with smaller trainable parameter ratios requires larger learning rates. For Random Masking with a very sparse mask, *e.g.*, 0.001% trainable parameters, the optimal learning rate can be as high as $1\mathrm{e}{-1}$, which is typically considered excessively large and unstable for standard NLP training. In fact, our experiments show that such an aggressive learning rate will lead to a fast divergence and a degraded performance for other baselines.

4.3. Investigations and Explanations

The results in Figure 3 raise two important questions about the selection of learning rates. The first one is why large learning rates do not diverge and work well for Random Masking. The second one is why small learning rates, which are suitable for full fine-tuning and traditional PEFT methods, do not work well for Random Masking. We provide the following empirical observations to explain the phenomena.

The Stability of Large Learning Rates: Sparser Random Masking Leads to a Flatter Loss Landscape. A well-known result in optimization theory says that gradient descent with a learning rate below $\Theta(1/L)$ is guaranteed to converge, where L is the smoothness coefficient given by the ℓ_2 norm of the hessian of the objective function (Bubeck et al., 2015). Therefore, the good performance of large learning rates suggests that the loss landscape after Random Masking is flat, *i.e.*, having a small Hessian norm.

We numerically calculate ℓ_2 norm of the hessian before and after training using the power method. The results in Figure 4(a) show that Random Masking leads to a smaller Hessian norm and thus a flatter loss landscape. Small Hessian norm also indicates that the loss landscape of PEFT is almost linear, which aligns with the findings in Malladi et al. (2023b).

Table 2: **The accuracy of Random Masking on image classification tasks.** The optimal learning rate are given in the parenthesis. These results show that the previous observations on the language domain also hold for the vision domain.

Method	**Params**	**CIFAR10**	**GTSRB**	**MNIST**	**SVHN**	**RESISC45**
FT	100%	98.5 (1e−5)	99.2 (1e−4)	99.8 (1e−5)	97.9 (1e−5)	96.7 (1e−5)
LoRA	0.3%	98.4 (1e−4)	99.2 (1e−3)	99.7 (1e−3)	97.8 (1e−3)	96.8 (1e−3)
Random Masking	1%	98.5 (1e−3)	99.2 (1e−3)	99.7 (1e−2)	97.8 (1e−3)	96.5 (1e−3)
Random Masking	0.1%	98.3 (1e−2)	98.9 (1e−2)	99.6 (1e−2)	97.3 (1e−2)	95.8 (1e−2)
Random Masking	0.01%	97.8 (1e−1)	96.8 (1e−1)	99.3 (1e−1)	95.6 (1e−1)	93.3 (1e−1)

Table 3: **The performance using full training split.** The results show that Random Masking is robust to the size of training set, and full-parameter fine-tuning performs better with a larger training set.

Model	**Task**	**FT**	**LoRA**	**Adapter**	**BitFit**	**Masking** (0.1%)	**Masking** (0.01%)	**Masking** (0.001%)
OPT-125m	SST-2	91.7	91.3	90.7	90.8	91.6	90.4	87.8
OPT-125m	MultiRC	69.1	70.0	69.4	69.0	69.5	68.3	68.1
OPT-1.3b	SST-2	95.1	95.8	95.6	95.4	95.4	95.4	94.8
OPT-1.3b	MultiRC	81.2	78.3	78.3	75.8	80.1	74.5	72.2

The Necessity of Large Learning Rates: Sparser Random Masking Leads to More Distant Solutions. Figure 3 shows that small learning rates work badly for sparse masking. Since these small learning rates are sufficient for convergence when the masking ratio is low, we attribute this failure to the underfitting of small learning rates. We validate this in Figure 4(b), which presents the performance on SST-2 dataset with longer training epochs and small learning rates. We observe that as the training epoch extends, the performance monotonically increases. Therefore, the failure of small learning rates is due to optimization rather than generalization, since they require a significantly large number of steps to fit the dataset.

The required number of steps can be reflected by the ℓ_2 distance between the initialization and final iterate. We show in Figure 4(b) that as the masking gets sparser, this distance becomes larger, even though only a smaller number of parameters are varied. This indicates that the iterates have to travel further to reach a minimizer.

Random Masking Demonstrates the Expressiveness of Pretrained LLMs. The above investigations reveals the following general picture: Random Masking deactivates a significant portion of dimensions, excluding minimizers that are easily reachable. However, thanks to the expressiveness of pretrained networks, there are still distant minimizers in the active dimensions, which require a larger learning rate to be effectively reached. Therefore, the success of Random Masking is not merely attributed to the method itself, but more to the underlying expressive power and generalization ability of pretrained LLMs. Random Masking serves as a tool to reveal the surprising expressive power of pretrained LLMs, which is a key message that we want to share with the community.

4.4. Ablations Studies and Additional Experiments

In this section, we provide further analyses to uncover how the task, the data size, the choice of base models, and the ways of selecting the mask affect the performance of Random Masking.

Fine-tuning Vision Models. To investigate the performance of Random Masking on vision tasks, we choose Clip ViT-B/16 as the pretrained model, and fine-tune it on 5 image classification tasks. The results are given in Table 2, which shows a close performance to full-parameter fine-tuning and a similar trend of optimal learning rate as in NLP tasks. The detailed setup are deferred to Appendix B.2.

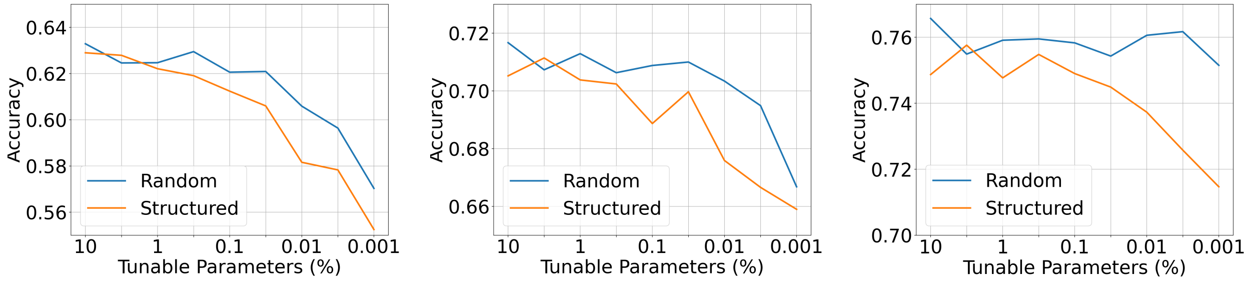
Varying Data Sizes. We demonstrate the robustness of Random Masking to the size of the training set. We choose the SST-2 and MultiRC datasets, which have 67.3k and 27.3k data points in the training split, respectively. We conduct full-dataset training on them, with the results presented in Table 3. The results indicate that the performance of Random Masking is consistent across different sizes of training set.

Furthermore, we observe that the influence of trainable parameter count is more evident in this full training set scenario. Notably, full-parameter fine-tuning performs comparably better than in the low data regime. This phenomenon is attributed to the pretrained model capacity relative to the training data, as larger datasets require more parameters to fit. This finding underscores again the critical role of expressive power in fine-tuning pretrained LLMs.

Varying Base Models. Next, we show that Random Masking is robust to the choice of pretrained models. We choose

Table 4: **The performance and optimal learning rates of Random Masking with Llama2 as the pretrained model.** The learning rates are searched from $\{1, 2, 5\}\times \{1e{-}1, 1e{-}2, 1e{-}3, 1e{-}4, 1e{-}5, 1e{-}6\}$. The results show that Random Masking is robust to the choice of base models.

Task	**FT**	**LoRA**	**Masking** (1%)	**Masking** (0.1%)	**Masking** (0.01%)	**Masking** (0.001%)
SST-2	94.7(1e−6)	95.4(1e−4)	95.4(1e−4)	95.5(1e−3)	95.5(1e−2)	95.5(5e−2)
WiC	71.9(1e−6)	72.7(1e−4)	72.1(2e−5)	70.6(5e−4)	70.5(1e−2)	71.7(5e−2)
RTE	85.9(1e−5)	86.5(1e−3)	85.4(1e−4)	85.4(1e−3)	85.6(1e−2)	83.0(5e−2)
COPA	87.0(1e−6)	85.0(1e−4)	87.0(2e−4)	87.0(2e−3)	88.0(5e−3)	88.0(5e−2)



OPT-125m

OPT-1.3b

OPT-13b

Figure 5: **Random Masking v.s. Structured Masking.** Structured Masking has a degraded and faster decaying performance. The complete results for Structured Masking can be found in Table 8 and Table 9.

Llama2-7b (Touvron et al., 2023b) as the pretrained model and conduct the experiments. The results are given in Table 4. We find that Compared with the OPT series model, Llama2 requires a more fine-grained learning rate search. The results indicate that the efficacy of Random Masking remains consistent across various pretrained base models, as long as the learning rate is properly selected.

Masks beyond Uniformly Random. Finally, we delve into the role of randomness in Random Masking. Randomly choosing the mask induces uniformity when the parameter count is large. To investigate its effect, we propose a contrary method which we call Structured Masking. Instead of randomly selecting the mask, Structured Masking chooses the trainable parameters along the columns of the weight matrix, as illustrated in 2(d).

The results for structured masking are presented in Figure 5. Compared with Random Masking, Structured Masking yields lower performance and exhibits a more rapid decline in accuracy as the trainable parameter count decreases. The performance gain of randomly selecting the mask indicates that the uniformity induced by randomness can be important for fine-tuning pretrained LLMs.

5. Theoretical Explanations

In this section, we uncover the interplay between Random Masking, loss landscape and learning rate by analyzing an overparameterized linear regression model. Our theoretical results show that for linear models, Random Masking can lead to a flatter landscape, a larger stable learning rate, and a more distant solution in the considered setup.

5.1. Setups

Consider fitting a linear model $f(\boldsymbol{w}) = \boldsymbol{w}^\top \boldsymbol{x}$ on a dataset $\{(\boldsymbol{x}_i, y_i)\}_{1\leq i\leq n}$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the feature vector and $y_i \in \mathbb{R}$ is the target. Let $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)^\top = (\boldsymbol{z}_1, \cdots, \boldsymbol{z}_d) \in \mathbb{R}^{n\times d}$ and $\boldsymbol{y} = (y_1, \cdots, y_n) \in \mathbb{R}^n$. We ignore the bias without loss of generality. Since pretrained model has a large parameter count, here we consider the overparameterized setting, *i.e.*, $d \gg n$.

To mimic the Random Masking method, we apply a random masking matrix on the feature vectors. We denote the random masking matrix as $\boldsymbol{M} := \mathrm{diag}(m_1, \cdots, m_d)$, where each m_i is sampled i.i.d. from $\mathrm{Binom}(p)$ and $p \in [0, 1]$ denotes the trainable parameter ratio. We denote $\tilde{\boldsymbol{w}}$ as the pretrained model weights, and $\boldsymbol{w}$ as the trainable weights in Random Masking.

We consider minimizing the following ℓ_2 loss using gradient descent with learning rate $\eta > 0$:

$$L(\boldsymbol{w}) = \frac{1}{2n}\|\boldsymbol{y} - \boldsymbol{X}(\tilde{\boldsymbol{w}} + \boldsymbol{M}\boldsymbol{w})\|^2. \tag{1}$$

Since $\boldsymbol{X}\tilde{\boldsymbol{w}}$ can be merged into $\boldsymbol{y}$, we assume without loss of generality that $\tilde{\boldsymbol{w}} = 0$. Denote the training trajectory as

$\{\boldsymbol{w}_i\}_{i\geq 0}$, where $\boldsymbol{w}_{i+1} = \boldsymbol{w}_i - \eta \nabla L(\boldsymbol{w}_i)$ and $\boldsymbol{w}_0 = 0$.

We use $\lambda_i(\boldsymbol{A})$ denote the i-th largest eigenvalue of matrix $\boldsymbol{A}$. When $\boldsymbol{A} = \boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{M}$, we drop the matrix and just use λ_i for brevity. Note that the smoothness of $L(\boldsymbol{w})$ is $\frac{1}{n}\lambda_1$.

5.2. Sparse Masking Leads to Small Eigenvalues

We first present the following concentration bound on the eigenvalues λ_i of matrix $\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{M}$.

Theorem 5.1. *Suppose that each entry of $\boldsymbol{X}$ is in $[0, r]$. Then for any $0 < \delta < 1$, with probability at least $1 - \delta$, the following inequality for λ_i holds for any i,*

$$|\lambda_i - p\lambda_i(\boldsymbol{X}^\top\boldsymbol{X})| \leq 2\sqrt{2dn^3r^4} + \sqrt{\frac{2\log(\frac{1}{\delta})}{dn^2r^4}}$$

The proofs are all deferred to Appendix A. This theorem shows that λ_i concentrates around $p\lambda_i(\boldsymbol{X}^\top\boldsymbol{X})$, which goes to zero as the trainable parameter ratio p goes to zero. Theorem 5.1 also contains a deviation term that scales like $O\left(\sqrt{d}\right)$, since we consider the overparameterized setting where $d \gg n$. Note that

$$\begin{aligned}\mathbb{E}\left(\sum_{1\leq i\leq n}\lambda_i\right) &= \mathbb{E}\operatorname{Tr}\left(\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{M}\right) = p\operatorname{Tr}\left(\boldsymbol{X}^\top\boldsymbol{X}\right)\\ &= p\|\boldsymbol{X}\|_F^2 = p\sum_{1\leq i\leq n}\sum_{1\leq j\leq d}x_{i,j}^2.\end{aligned}$$

Therefore, $\mathbb{E}\left(\sum_i \lambda_i\right)$ scales like $\Theta(d) \gg O\left(\sqrt{d}\right)$ under some mild conditions on the feature distribution. This indicates that despite having a deviation term, Theorem 5.1 characterizes the sharp concentration of λ_i in the overparameterized setup.

5.3. Analysis of Gradient Descent Trajectories

Next, we show that the optimization property of this problem has a crucial dependency on the spectrum λ_i of the matrix $\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{M}$.

The following proposition is standard in optimization literature, which shows that the maximal stable learning rate is determined by the largest singular value.

Proposition 5.2. *The training trajectory $\{\boldsymbol{w}_i\}_{i\geq 0}$ converges for any initialization if and only if $\eta < \frac{2n}{\lambda_1}$.*

Combined with Theorem 5.1, we know that the learning rate η can be large as the trainable parameter number becomes smaller.

Denote $\hat{\boldsymbol{w}}$ as the convergence point of GD, if learning rate η satisfy the bound in Proposition 5.2. The following proposition gives a lower bound on the norm of $\hat{\boldsymbol{w}}$.

Proposition 5.3. *Suppose each y_i is generated using a ground truth weight vector $\boldsymbol{w}^*$ and i.i.d. Gaussian random noise ϵ_i with variance σ^2,* i.e. *$y_i = \boldsymbol{w}^{*,\top}\boldsymbol{x}_i + \epsilon_i$. Suppose that $\eta < \frac{2n}{\lambda_1}$. Then the expected norm of $\hat{w}$ can be bounded as*

$$\mathbb{E}\left[\|\hat{\boldsymbol{w}}\|^2 \mid \boldsymbol{M}\right] \geq \sum_{i:\lambda_i>0}\frac{\sigma^2}{\lambda_i}, \tag{2}$$

where the expectation is taken over the randomness of noise $\epsilon_i, 1 \leq i \leq n$.

This proposition together with Theorem 5.1 shows that as the trainable parameter ratio p gets smaller, gradient descent will converge to a more distant solution, which aligns with our empirical findings. Note that $\boldsymbol{w}^*$ can encompass the pretrained weights $\tilde{\boldsymbol{w}}$.

6. Conclusions and Discussions

This paper shows that a randomly masked LLM can be successfully fine-tuned on standard NLP benchmarks, as long as the learning rate is properly set. Our experiments show that Random Masking achieves comparable performance with other PEFT algorithms, despite having a simple algorithm design and a reduced amount of trainable parameters. We investigate its mechanism both empirically and theoretically, and demonstrate that the large expressive power of pretrained models and a benign loss landscape are the underlying factors for its success. Overall, our findings illuminate the under-explored potential of pretrained models and suggest that PEFT can stay effective with much fewer trainable parameters and simpler algorithmic designs.

Our research suggests several promising directions for future exploration.

Firstly, while Random Masking has demonstrated success, it should not be regarded as a state-of-the-art PEFT algorithm, but rather as a tool to reveal the huge expressiveness of pretrained models. Consequently, Random Masking may encounter challenges with complex fine-tuning tasks that requires larger expressive power. We leave this for future investigations.

Secondly, our results show that pretraining and fine-tuning may require different optimization algorithms. The different task difficulty and loss landscape property in these two phases suggest a need for novel optimization algorithms specifically tailored for fine-tuning smaller-scale modules on large-scale pretrained models.

Thirdly, Random Masking has a deep connection to neural network pruning, and we anticipate its success in fine-tuning LLMs will catalyze further research in this related field.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Aghajanyan, A., Zettlemoyer, L., and Gupta, S. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

Bentivogli, L., Clark, P., Dagan, I., and Giampiccolo, D. The fifth pascal recognizing textual entailment challenge. *TAC*, 7(8):1, 2009.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Bubeck, S. et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Chen, J., Zhang, A., Shi, X., Li, M., Smola, A., and Yang, D. Parameter-efficient fine-tuning design spaces. *arXiv preprint arXiv:2301.01821*, 2023.

Cheng, G., Han, J., and Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, Oct 2017. ISSN 1558-2256. doi: 10.1109/jproc.2017.2675998. URL http://dx.doi.org/10.1109/JPROC.2017.2675998.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Dagan, I., Glickman, O., and Magnini, B. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pp. 177–190. Springer, 2005.

De Marneffe, M.-C., Simons, M., and Tonhauser, J. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pp. 107–124, 2019.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., et al. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*, 2022.

Ding, N., Lv, X., Wang, Q., Chen, Y., Zhou, B., Liu, Z., and Sun, M. Sparse low-rank adaptation of pre-trained language models. *arXiv preprint arXiv:2311.11696*, 2023.

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., and Gardner, M. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*, 2019.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Gale, T., Zaharia, M., Young, C., and Elsen, E. Sparse GPU kernels for deep learning. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2020*, 2020.

Gao, P., Han, J., Zhang, R., Lin, Z., Geng, S., Zhou, A., Zhang, W., Lu, P., He, C., Yue, X., et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023.

Giampiccolo, D., Magnini, B., Dagan, I., and Dolan, W. B. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pp. 1–9, 2007.

Guo, D., Rush, A. M., and Kim, Y. Parameter-efficient transfer learning with diff pruning. *arXiv preprint arXiv:2012.07463*, 2020.

Haim, R. B., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., and Szpektor, I. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, volume 7, pp. 785–794, 2006.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

Hao, Y., Dong, L., Wei, F., and Xu, K. Visualizing and understanding the effectiveness of bert. *arXiv preprint arXiv:1908.05620*, 2019.

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., and Neubig, G. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021a.

He, R., Liu, L., Ye, H., Tan, Q., Ding, B., Cheng, L., Low, J.-W., Bing, L., and Si, L. On the effectiveness of adapter-based tuning for pretrained language model adaptation. *arXiv preprint arXiv:2106.03164*, 2021b.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Hu, S., Zhang, Z., Ding, N., Wang, Y., Wang, Y., Liu, Z., and Sun, M. Sparse structure search for delta tuning. *Advances in Neural Information Processing Systems*, 35: 9853–9865, 2022.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Jaiswal, A. K., Ma, H., Chen, T., Ding, Y., and Wang, Z. Training your sparse neural network better with any mask. In *International Conference on Machine Learning*, pp. 9833–9844. PMLR, 2022.

Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N. Visual prompt tuning. In *European Conference on Computer Vision*, pp. 709–727. Springer, 2022.

Jie, S. and Deng, Z.-H. Convolutional bypasses are better vision transformer adapters. *arXiv preprint arXiv:2207.07039*, 2022.

Karimi Mahabadi, R., Henderson, J., and Ruder, S. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34: 1022–1035, 2021.

Khashabi, D., Chaturvedi, S., Roth, M., Upadhyay, S., and Roth, D. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. 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. 252–262, 2018.

Koohpayegani, S. A., Navaneet, K., Nooralinejad, P., Kolouri, S., and Pirsiavash, H. Nola: Networks as linear combination of low rank random basis. *arXiv preprint arXiv:2310.02556*, 2023.

Kopiczko, D. J., Blankevoort, T., and Asano, Y. M. Vera: Vector-based random matrix adaptation. *arXiv preprint arXiv:2310.11454*, 2023.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

LeCun, Y. and Cortes, C. The mnist database of handwritten digits. 2005. URL https://api.semanticscholar.org/CorpusID:60282629.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Levesque, H., Davis, E., and Morgenstern, L. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Li, Y., Luo, F., Tan, C., Wang, M., Huang, S., Li, S., and Bai, J. Parameter-efficient sparsity for large language models fine-tuning. *arXiv preprint arXiv:2205.11005*, 2022.

Lialin, V., Deshpande, V., and Rumshisky, A. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*, 2023.

Lin, Y.-B., Sung, Y.-L., Lei, J., Bansal, M., and Bertasius, G. Vision transformers are parameter-efficient audio-visual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2299–2309, 2023.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. A. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35: 1950–1965, 2022.

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., and Tang, J. Gpt understands, too. *AI Open*, 2023.

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pp. 2736–2744, 2017.

Lv, K., Yang, Y., Liu, T., Gao, Q., Guo, Q., and Qiu, X. Full parameter fine-tuning for large language models with limited resources. *arXiv preprint arXiv:2306.09782*, 2023.

Malladi, S., Gao, T., Nichani, E., Damian, A., Lee, J. D., Chen, D., and Arora, S. Fine-tuning language models with just forward passes. *arXiv preprint arXiv:2305.17333*, 2023a.

Malladi, S., Wettig, A., Yu, D., Chen, D., and Arora, S. A kernel-based view of language model fine-tuning. In *International Conference on Machine Learning*, pp. 23610–23641. PMLR, 2023b.

Mao, Y., Mathias, L., Hou, R., Almahairi, A., Ma, H., Han, J., Yih, W.-t., and Khabsa, M. Unipelt: A unified framework for parameter-efficient language model tuning. *arXiv preprint arXiv:2110.07577*, 2021.

Molchanov, D., Ashukha, A., and Vetrov, D. Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning*, pp. 2498–2507. PMLR, 2017.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A. Y., et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 7. Granada, Spain, 2011.

Nikdan, M., Tabesh, S., Crnčević, E., and Alistarh, D. Rosa: Accurate parameter-efficient fine-tuning via robust adaptation. *arXiv preprint arXiv:2401.04679*, 2024.

Ortiz-Jimenez, G., Favero, A., and Frossard, P. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 36, 2024.

Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., and Gurevych, I. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020.

Pilehvar, M. T. and Camacho-Collados, J. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. *arXiv preprint arXiv:1808.09121*, 2018.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Roemmele, M., Bejan, C. A., and Gordon, A. S. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*, 2011.

Rücklé, A., Geigle, G., Glockner, M., Beck, T., Pfeiffer, J., Reimers, N., and Gurevych, I. Adapterdrop: On the efficiency of adapters in transformers. *arXiv preprint arXiv:2010.11918*, 2020.

Shi, E., Wang, Y., Zhang, H., Du, L., Han, S., Zhang, D., and Sun, H. Towards efficient fine-tuning of pre-trained code models: An experimental study and beyond. *arXiv preprint arXiv:2304.05216*, 2023.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pp. 1453–1460. IEEE, 2011.

Su, Y., Chan, C.-M., Cheng, J., Qin, Y., Lin, Y., Hu, S., Yang, Z., Ding, N., Liu, Z., and Sun, M. Arbitrary few parameters are good enough for adapting large-scale pre-trained language models. *arXiv preprint arXiv:2306.02320*, 2023a.

Su, Y., Chan, C.-M., Cheng, J., Qin, Y., Lin, Y., Hu, S., Yang, Z., Ding, N., Sun, X., Xie, G., et al. Exploring the impact of model scaling on parameter-efficient tuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15062–15078, 2023b.

Sung, Y.-L., Nair, V., and Raffel, C. A. Training neural networks with fixed sparse masks. *Advances in Neural Information Processing Systems*, 34:24193–24205, 2021.

Sung, Y.-L., Cho, J., and Bansal, M. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5227–5237, 2022.

Tao, T. *Topics in random matrix theory*, volume 132. American Mathematical Society, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

Xu, R., Luo, F., Zhang, Z., Tan, C., Chang, B., Huang, S., and Huang, F. Raise a child in large language model: Towards effective and generalizable fine-tuning. *arXiv preprint arXiv:2109.05687*, 2021.

Xu, Y., Xie, L., Gu, X., Chen, X., Chang, H., Zhang, H., Chen, Z., Zhang, X., and Tian, Q. Qa-lora: Quantization-aware low-rank adaptation of large language models. *arXiv preprint arXiv:2309.14717*, 2023.

Zaken, E. B., Ravfogel, S., and Goldberg, Y. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.

Zelikman, E., Huang, Q., Liang, P., Haber, N., and Goodman, N. D. Just one byte (per gradient): A note on low-bandwidth decentralized language model finetuning using shared randomness. *arXiv preprint arXiv:2306.10015*, 2023.

Zhang, F., Li, L., Chen, J., Jiang, Z., Wang, B., and Qian, Y. Increlora: Incremental parameter allocation method for parameter-efficient fine-tuning. *arXiv preprint arXiv:2308.12043*, 2023a.

Zhang, L., Zhang, L., Shi, S., Chu, X., and Li, B. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303*, 2023b.

Zhang, M., Shen, C., Yang, Z., Ou, L., Yu, X., Zhuang, B., et al. Pruning meets low-rank parameter-efficient fine-tuning. *arXiv preprint arXiv:2305.18403*, 2023c.

Zhang, Q., Chen, M., Bukharin, A., He, P., Cheng, Y., Chen, W., and Zhao, T. Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023d.

Zhang, R., Han, J., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., Gao, P., and Qiao, Y. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023e.

Zhang, S., Liu, X., Liu, J., Gao, J., Duh, K., and Van Durme, B. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*, 2018.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Zhao, M., Lin, T., Mi, F., Jaggi, M., and Schütze, H. Masking as an efficient alternative to finetuning for pretrained language models. *arXiv preprint arXiv:2004.12406*, 2020.

Zhou, H., Wan, X., Vulić, I., and Korhonen, A. Autopeft: Automatic configuration search for parameter-efficient fine-tuning. *arXiv preprint arXiv:2301.12132*, 2023.

Zhou, Y. and Srikumar, V. A closer look at how fine-tuning changes bert. *arXiv preprint arXiv:2106.14282*, 2021.

Zhu, Y., Feng, J., Zhao, C., Wang, M., and Li, L. Serial or parallel? plug-able adapter for multilingual machine translation. *arXiv preprint arXiv:2104.08154*, 6(3), 2021.

Zi, B., Qi, X., Wang, L., Wang, J., Wong, K.-F., and Zhang, L. Delta-lora: Fine-tuning high-rank parameters with the delta of low-rank matrices. *arXiv preprint arXiv:2309.02411*, 2023.

Appendix

A. Proofs

A.1. Proof for Proposition 5.2

Proof. This proposition is a standard result from convex optimization, and we include the proof here for completeness. First we prove by induction that the optimization trajectory $\{\boldsymbol{w}_t\}$ has the following closed form:

$$\boldsymbol{w}_t = \left(\boldsymbol{I} - \frac{\eta}{n}\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{M}\right)^t \left(\boldsymbol{w}_0 - (\boldsymbol{X}\boldsymbol{M})^\dagger\boldsymbol{y}\right) + (\boldsymbol{X}\boldsymbol{M})^\dagger\boldsymbol{y}, \tag{3}$$

where $(\boldsymbol{X}\boldsymbol{M})^\dagger$ denotes the pseudo-inverse of matrix $\boldsymbol{X}\boldsymbol{M}$. Equation 3 holds for $t = 0$. Suppose it holds for $t = k$, then for $t = k + 1$, we have

$$\begin{aligned}
\boldsymbol{w}_{t+1} &= \boldsymbol{w}_t - \eta\nabla L(\boldsymbol{w}_t)\\
&= \boldsymbol{w}_t - \frac{\eta}{n}\left(\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{M}\boldsymbol{w}_t - \boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{y}\right)\\
&= \left(\boldsymbol{I} - \frac{\eta}{n}\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{M}\right)\left[\left(\boldsymbol{I} - \frac{\eta}{n}\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{M}\right)^t \left(\boldsymbol{w}_0 - (\boldsymbol{X}\boldsymbol{M})^\dagger\boldsymbol{y}\right) + (\boldsymbol{X}\boldsymbol{M})^\dagger\boldsymbol{y}\right] - \frac{\eta}{n}\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{y}\\
&= \left(\boldsymbol{I} - \frac{\eta}{n}\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{M}\right)^{t+1} \left(\boldsymbol{w}_0 - (\boldsymbol{X}\boldsymbol{M})^\dagger\boldsymbol{y}\right) + (\boldsymbol{X}\boldsymbol{M})^\dagger\boldsymbol{y},
\end{aligned}$$

which completes the proof for Equation 3.

Recall that λ_1 is the largest eigenvalue of $\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{M}$. If $\eta < \frac{2n}{\lambda_1}$, then $\left\|\left(\boldsymbol{I} - \frac{\eta}{n}\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{M}\right)^{t+1}\right\|_2 < 1$, and therefore $\boldsymbol{w}_t$ converges to $(\boldsymbol{X}\boldsymbol{M})^\dagger y$.

On the other hand, if $\eta \geq \frac{2n}{\lambda_1}$, then we can denote $\boldsymbol{v}_1$ as the eigen-vector corresponding to λ_1, and choose $\boldsymbol{w}_0 = (\boldsymbol{X}\boldsymbol{M})^\dagger\boldsymbol{y}+\boldsymbol{v}$. In this case,

$$\boldsymbol{w}_t = (1 - \frac{\eta}{n}\lambda_1)^{t+1}\boldsymbol{v}_1 + (\boldsymbol{X}\boldsymbol{M})^\dagger\boldsymbol{y},$$

which does not converge. □

A.2. Proof for Proposition 5.3

Proof. Let $\boldsymbol{\varepsilon} = (\varepsilon_1, \cdots, \varepsilon_n)$. From the proof of Proposition 5.2, we know that

$$\hat{\boldsymbol{w}} = (\boldsymbol{X}\boldsymbol{M})^\dagger\boldsymbol{y} = (\boldsymbol{X}\boldsymbol{M})^\dagger\left(\boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\varepsilon}\right).$$

Therefore,

$$\begin{aligned}
\mathbb{E}\left\|\hat{\boldsymbol{w}}\right\|^2 &= \mathbb{E}\left[\left(\boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\varepsilon}\right)^\top (\boldsymbol{X}\boldsymbol{M})^{\dagger,\top}(\boldsymbol{X}\boldsymbol{M})^\dagger\left(\boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\varepsilon}\right)\right]\\
&= \mathbb{E}\left[\boldsymbol{\varepsilon}^\top(\boldsymbol{X}\boldsymbol{M})^{\dagger,\top}(\boldsymbol{X}\boldsymbol{M})^\dagger\boldsymbol{\varepsilon}\right] + \left(\boldsymbol{X}\boldsymbol{w}^*\right)^\top (\boldsymbol{X}\boldsymbol{M})^{\dagger,\top}(\boldsymbol{X}\boldsymbol{M})^\dagger\left(\boldsymbol{X}\boldsymbol{w}^*\right)\\
&\geq \mathbb{E}\left[\boldsymbol{\varepsilon}^\top(\boldsymbol{X}\boldsymbol{M})^{\dagger,\top}(\boldsymbol{X}\boldsymbol{M})^\dagger\boldsymbol{\varepsilon}\right]\\
&= \text{Tr}\left[(\boldsymbol{X}\boldsymbol{M})^{\dagger,\top}(\boldsymbol{X}\boldsymbol{M})^\dagger\mathbb{E}\left(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top\right)\right]\\
&= \sigma^2\text{Tr}\left[(\boldsymbol{X}\boldsymbol{M})^{\dagger,\top}(\boldsymbol{X}\boldsymbol{M})^\dagger\right]\\
&= \sigma^2\text{Tr}\left[(\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{M})^\dagger\right]\\
&= \sum_{i:\lambda_i>0}\frac{\sigma^2}{\lambda_i}
\end{aligned}$$

□

A.3. Proof for Theorem 5.1

We first prove the following lemma.

Lemma A.1. *Let $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^\top - p\boldsymbol{X}\boldsymbol{X}^\top$. For any $\boldsymbol{u} \in \mathbb{R}^n$, the moment generating function of $\langle \boldsymbol{u}, \boldsymbol{Q}\boldsymbol{u}\rangle$ can be bounded as*

$$\mathbb{E}\exp\left(t\left\langle \boldsymbol{u}, \boldsymbol{Q}\boldsymbol{u}\right\rangle\right) \le \exp\left(\sum_{i=1}^{d} \frac{t^2(\boldsymbol{z}_i^\top \boldsymbol{u})^4}{8}\right) \le \exp\left(\frac{dn^2 r^4 \|\boldsymbol{u}\|^4 t^2}{8}\right)$$

Proof. By the independence of each m_i, we can simplify the moment generating function as

$$\begin{aligned}\mathbb{E}\exp\left(t\left\langle \boldsymbol{u}, \boldsymbol{Q}\boldsymbol{u}\right\rangle\right) &= \mathbb{E}\exp\left(t\sum_{i=1}^{d}\left\langle \boldsymbol{u}, (m_i - p)\boldsymbol{z}_i\boldsymbol{z}_i^\top \boldsymbol{u}\right\rangle\right)\\ &= \mathbb{E}\exp\left(t\sum_{i=1}^{d}(m_i - p)\left(\boldsymbol{z}_i^\top \boldsymbol{u}\right)^2\right)\\ &= \Pi_{i=1}^{d}\mathbb{E}\exp\left(t(m_i - p)\left(\boldsymbol{z}_i^\top \boldsymbol{u}\right)^2\right)\end{aligned}$$

Note that $(m_i - p)\left(\boldsymbol{z}_i^\top \boldsymbol{u}\right)^2$ as zero mean and is bounded in $\left[-p\left(\boldsymbol{z}_i^\top \boldsymbol{u}\right)^2, (1-p)\left(\boldsymbol{z}_i^\top \boldsymbol{u}\right)^2\right]$. According to Example 2.4 in Wainwright (2019), the random variable is sub-Gaussian with parameter $\frac{1}{2}\left[(1-p)\left(\boldsymbol{z}_i^\top \boldsymbol{u}\right)^2 + p\left(\boldsymbol{z}_i^\top \boldsymbol{u}\right)^2\right] = \frac{1}{2}\left(\boldsymbol{z}_i^\top \boldsymbol{u}\right)^2$. This implies that

$$\mathbb{E}\exp\left(t(m_i - p)\left(\boldsymbol{z}_i^\top \boldsymbol{u}\right)^2\right) \le \exp\left(\frac{t^2(\boldsymbol{z}_i^\top \boldsymbol{u})^4}{8}\right),$$

which proves the first inequality of this lemma. The second inequality is immediate by Cauchy-Schwarz inequality and the assumption that the entries of $\boldsymbol{z}_i$ are bounded by r. □

Next we prove Theorem 5.1.

Proof. From the discretization argument in the proof of Theorem 6.5 in Wainwright (2019), we know that there exists $N \le 17^n$, and unit vectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_N \in \mathbb{S}^{n-1}$, such that

$$\|\boldsymbol{Q}\|_2 \le 2\max_{1\le j\le N} |\left\langle \boldsymbol{v}_j, \boldsymbol{Q}\boldsymbol{v}_j\right\rangle|.$$

Therefore, the moment generating function of $\|\boldsymbol{Q}\|_2$ can be bounded as

$$\begin{aligned}\mathbb{E}\left[\exp\left(\lambda\|\boldsymbol{Q}\|_2\right)\right] &\le \mathbb{E}\left[\exp\left(2\lambda\max_{1\le j\le N}|\left\langle \boldsymbol{v}_j, \boldsymbol{Q}\boldsymbol{v}_j\right\rangle|\right)\right]\\ &\le \sum_{j=1}^{N}\left\{\mathbb{E}\left[\exp\left(2\lambda\left\langle \boldsymbol{v}_j, \boldsymbol{Q}\boldsymbol{v}_j\right\rangle\right)\right] + \mathbb{E}\left[\exp\left(-2\lambda\left\langle \boldsymbol{v}_j, \boldsymbol{Q}\boldsymbol{v}_j\right\rangle\right)\right]\right\}\end{aligned}$$

According to Lemma A.1, it can be further bounded as

$$\mathbb{E}\left[\exp\left(\lambda\|\boldsymbol{Q}\|_2\right)\right] \le 2N\exp\left(\frac{dn^2r^4\lambda^2}{2}\right) \le \exp\left(4n + \frac{dn^2r^4\lambda^2}{2}\right),$$

where in the last inequality we use the fact that $2 \times 17^n \le \exp(4n)$ for $n \ge 1$.

The bound on moment generating function implies that for any $t, \lambda > 0$,

$$\begin{aligned}\Pr\left(\|\boldsymbol{Q}\|_2 > t\right) &\le \Pr\left(\exp(\lambda\|\boldsymbol{Q}\|_2) \ge \exp(\lambda t)\right)\\ &\le \frac{\mathbb{E}\exp\left(\lambda\|\boldsymbol{Q}\|_2\right)}{\exp(\lambda t)}\\ &\le \exp\left(4n + \frac{dn^2r^4\lambda^2}{2} - \lambda t\right).\end{aligned}$$

Taking $\lambda = \frac{t}{dn^2r^4}$, we get

$$\Pr\left(\|\boldsymbol{Q}\|_2 > t\right) \leq \exp\left(4n - \frac{t^2}{2dn^2r^4}\right).$$

Replace t with $t + 2\sqrt{2dn^3r^4}$, we have

$$\Pr\left(\|\boldsymbol{Q}\|_2 > t + 2\sqrt{2dn^3r^4}\right) \leq \exp\left(-\frac{t^2}{2dn^2r^4}\right).$$

Therefore, for any $0 < \delta < 1$, we know that with probability at least $1 - \delta$, we have

$$\|\boldsymbol{Q}\|_2 \leq 2\sqrt{2dn^3r^4} + \sqrt{\frac{2\log(\frac{1}{\delta})}{dn^2r^4}}.$$

This inequality together with Weyl's theorem (*e.g.*, see Equation 1.54 in Tao (2023)) implies that

$$|\lambda_i - p\lambda_i(\boldsymbol{X}^\top\boldsymbol{X})| = |\lambda_i(\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^\top) - \lambda_i(p\boldsymbol{X}\boldsymbol{X}^\top)| \leq \|\boldsymbol{Q}\|_2 \leq 2\sqrt{2dn^3r^4} + \sqrt{\frac{2\log(\frac{1}{\delta})}{dn^2r^4}}$$

□

B. Additional Experiment Results

B.1. Details of Random Masking Experiments

Details of Datasets. The SuperGLUE benchmark consists 8 natural language understanding tasks, including BoolQ (Clark et al., 2019), CB (De Marneffe et al., 2019), COPA (Roemmele et al., 2011), MultiRC (Khashabi et al., 2018), ReCoRD (Zhang et al., 2018), RTE (Dagan et al., 2005; Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009), WiC (Pilehvar & Camacho-Collados, 2018), WSC (Levesque et al., 2012). We also include SST-2 dataset (Socher et al., 2013) and two language generation tasks SQuAD (Rajpurkar et al., 2016) and DROP (Dua et al., 2019).

Training Procedures. We choose the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.999, \varepsilon = 1\text{e}{-8}$. We perform a grid search of learning rate from $\{1\text{e}{-1}, 1\text{e}{-2}, 1\text{e}{-3}, 1\text{e}{-4}, 1\text{e}{-5}, 1\text{e}{-6}\}$. We follow the practice of Malladi et al. (2023a) and Dettmers et al. (2023), and use a constant learning rate schedule. The number of training epochs is set to 5. The batch size is set to 8 per GPU. We run each experiment three times and report the average metrics.

Details for Additional Baselines. We run experiments on additional baselines including Adapter, Prefix-Tuning, BitFit and AdaLoRA. For Adapter, we use the original design of adapters in Houlsby et al. (2019) and set the bottleneck width to 8. For Prefix-Tuning, we set the number of virtual tokens to 5 and initialize the prefix with the activations of real words. We choose $r = 8$ and $\alpha = 16$ for AdaLoRA, and apply it only to the query and value matrix in each attention layer. BitFit is applied to all the bias vectors in the network.

B.2. Details for Experiments in the Vision Domain

We choose ViT-B/16(Radford et al., 2021) as the pretrained model, and perform image classification tasks by fine-tuning on the following 5 datasets: CIFAR10 (Krizhevsky et al., 2009), GTSRB (Stallkamp et al., 2011), MNIST (LeCun & Cortes, 2005), SVHN (Netzer et al., 2011), RESISC45 (Cheng et al., 2017). We follow the setup of Ilharco et al. (2022) and Ortiz-Jimenez et al. (2024), which fix the classification head for each task. The model is fine-tuned for 2000 steps with a batch size of 128. We choose The AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.999, \varepsilon = 1\text{e}{-8}$ and weight decay of 0.1. The learning rate is searched from $\{1\text{e}{-1}, 1\text{e}{-2}, 1\text{e}{-3}, 1\text{e}{-4}, 1\text{e}{-5}, 1\text{e}{-6}\}$. We use cosine annealing learning rate schedule with 200 warm-up steps. To show the robustness of Random Masking to the selection of target modules, we apply Random Masking to the MLP layers, rather than the attention layers in NLP tasks. The trainable parameter ratio for Random Masking is selected from $\{1\%, 0.1\%, 0.01\%\}$. We choose full-parameter tuning and LoRA as baselines. We apply LoRA to the MLP layers, with $r = 8$ and $\alpha = 16$.

B.3. Complete Experiment Results for Random Masking

We provide the complete random masking experiment results with different trainable parameter ratio in Table 5. The optimal learning rates of Random Masking and baselines are provided in Table 6 and Table 7. The plot for Random Masking with different learning rates are provided in Figure 6, 7 and 8. The analog of Figure 4 on different datasets are given in Figure 9, 10 and 11. The complete results of Structured Masking are provided in Table 8 and Table 9.

Table 5: **Random Masking achieves comparable test accuracy with fewer trainable parameters (complete results).** This table displays the test performance of different methods. Here, FT stands for full parameter fine-tuning, Prefix stands for Prefix-Tuning, Masking stands for Random Masking. Params stands for the trainable parameter ratio, which is the number of trainable parameters divided by the total parameter count of the original pretrained models.

Model	Method	Params	SST-2	RTE	WSC	WiC	CB	BoolQ	MultiRC	COPA	ReCoRD	SQuAD	DROP	Avg
OPT-125m	FT	100%	88.1	63.5	63.5	60.3	81.0	62.9	64.7	66.0	50.8	62.4	22.8	62.36
	Adapter	0.265%	86.0	62.3	63.5	60.4	69.0	62.7	65.0	68.0	51.3	61.5	21.9	61.07
	LoRA	0.235%	86.5	59.9	63.5	59.6	82.1	63.6	64.2	67.3	51.2	62.9	21.6	62.04
	AdaLoRA	0.235%	87.7	62.3	63.5	59.1	69.6	63.2	63.5	69.3	51.2	63.4	24.2	61.57
	Prefix	0.074%	88.1	58.5	63.5	58.0	69.6	63.9	62.2	64.7	50.6	59.7	20.1	59.91
	BitFit	0.066%	86.5	60.0	63.5	59.8	70.8	62.7	64.7	69.7	51.0	59.9	21.6	60.93
	Masking	10%	87.3	63.7	63.5	60.8	89.9	64.2	63.4	67.0	51.3	62.3	22.9	63.29
	Masking	5%	87.0	65.7	63.5	60.7	78.6	63.7	63.5	67.0	51.4	63.3	22.7	62.46
	Masking	1%	87.6	62.1	63.1	60.2	81.5	63.9	64.8	67.0	51.4	62.3	23.3	62.47
	Masking	0.5%	87.0	64.4	63.5	60.8	84.5	63.7	65.6	67.0	51.3	62.0	22.7	62.95
	Masking	0.1%	87.3	60.8	62.2	60.2	82.7	63.6	63.1	67.3	51.3	61.6	22.6	62.06
	Masking	0.05%	86.9	61.6	63.5	60.8	84.5	63.7	62.6	66.7	51.3	59.6	21.9	62.09
	Masking	0.01%	86.1	59.1	63.5	60.3	73.8	62.9	63.4	68.3	51.4	55.7	22.1	60.59
	Masking	0.005%	86.9	57.9	64.4	59.5	74.4	63.8	58.5	67.0	51.5	53.1	19.2	59.64
	Masking	0.001%	84.7	56.1	60.3	55.8	70.8	61.6	59.5	69.3	51.2	41.9	16.1	57.03
OPT-1.3b	FT	100%	93.7	70.5	63.1	62.7	85.7	69.5	67.6	76.7	71.8	81.2	29.3	70.16
	Adapter	0.134%	93.3	73.5	62.5	60.9	89.9	70.1	69.1	75.0	71.6	81.8	31.0	70.79
	LoRA	0.120%	93.4	72.6	63.5	65.5	78.6	71.4	69.9	81.0	71.2	82.1	29.9	70.81
	AdaLoRA	0.120%	93.9	74.1	62.2	61.8	79.2	71.0	67.6	76.0	71.0	81.1	31.2	69.91
	Prefix	0.037%	93.2	75.1	59.3	62.0	77.4	73.3	68.6	80.0	70.8	80.9	30.0	70.06
	BitFit	0.034%	93.0	72.0	63.5	62.9	86.9	71.7	67.9	76.7	71.8	82.0	29.2	70.69
	Masking	10%	93.4	72.6	63.5	66.2	91.1	73.2	68.7	76.0	71.4	82.0	30.4	71.67
	Masking	5%	93.8	71.0	63.5	63.2	88.1	71.7	68.3	76.3	71.9	81.5	28.9	70.73
	Masking	1%	93.5	70.5	63.5	64.8	89.9	71.9	69.3	75.7	71.5	82.4	31.2	71.29
	Masking	0.5%	93.7	70.2	63.5	61.1	89.3	71.9	68.1	76.3	71.8	81.7	29.3	70.63
	Masking	0.1%	93.3	72.7	63.8	62.3	89.9	71.5	68.3	75.3	71.7	81.1	29.7	70.88
	Masking	0.05%	93.3	73.9	63.1	63.4	86.3	71.3	69.0	76.3	72.0	81.4	30.9	71.00
	Masking	0.01%	92.6	70.0	63.5	62.7	82.1	71.5	68.8	77.7	71.5	81.4	31.9	70.34
	Masking	0.005%	92.6	70.9	63.5	59.4	81.5	70.3	68.8	76.0	72.0	80.7	28.7	69.49
	Masking	0.001%	92.7	65.0	63.5	60.4	74.4	67.1	59.0	74.3	71.0	77.6	28.5	66.68
OPT-13b	FT	100%	94.9	81.1	62.5	65.4	81.0	79.8	76.1	89.3	81.3	87.3	35.3	75.82
	Adapter	0.057%	95.3	83.6	58.3	68.2	91.1	80.6	69.0	88.3	80.9	88.0	34.8	76.19
	LoRA	0.051%	95.0	83.8	63.5	65.2	79.8	81.3	73.2	88.0	81.4	88.6	34.7	75.86
	AdaLoRA	0.051%	95.0	84.5	63.5	67.3	81.0	81.7	70.7	89.0	81.7	87.5	37.9	76.35
	Prefix	0.016%	94.5	82.8	61.9	66.0	85.7	81.4	73.7	89.3	81.9	87.4	34.5	76.28
	BitFit	0.014%	95.1	82.9	63.5	64.9	91.7	80.3	74.9	87.0	80.9	88.2	33.9	76.66
	Masking	10%	95.3	81.3	62.8	66.8	89.3	79.2	73.3	89.0	81.6	88.4	35.3	76.57
	Masking	5%	94.9	81.5	63.5	67.5	76.2	81.1	72.1	88.7	81.5	88.7	34.7	75.49
	Masking	1%	95.0	81.2	62.8	66.3	85.1	78.7	72.2	89.0	81.6	88.2	34.9	75.91
	Masking	0.5%	95.1	81.2	64.7	66.7	83.3	80.6	71.7	88.3	81.6	87.4	34.8	75.95
	Masking	0.1%	95.1	80.6	59.6	65.5	84.5	79.6	75.8	89.3	81.6	88.1	34.4	75.83
	Masking	0.05%	95.0	81.1	63.1	66.5	81.5	80.7	70.1	87.7	81.5	87.8	34.7	75.43
	Masking	0.01%	94.8	82.7	59.9	66.0	88.7	79.7	73.4	87.0	81.6	87.6	35.3	76.06
	Masking	0.005%	95.1	82.9	63.8	66.4	85.1	80.4	72.5	87.7	81.5	87.3	35.2	76.17
	Masking	0.001%	95.1	80.1	60.6	65.4	85.7	78.7	73.2	87.7	81.6	86.0	32.6	75.15

Table 6: **The optimal learning rate of Random Masking**, which are obtained via grid search in {1e−1, 1e−2, 1e−3, 1e−4, 1e−5, 1e−6}. Here, Masking stands for Random Masking. This table shows that **the optimal learning rate has a negative relationship with the number of trainable parameters.**

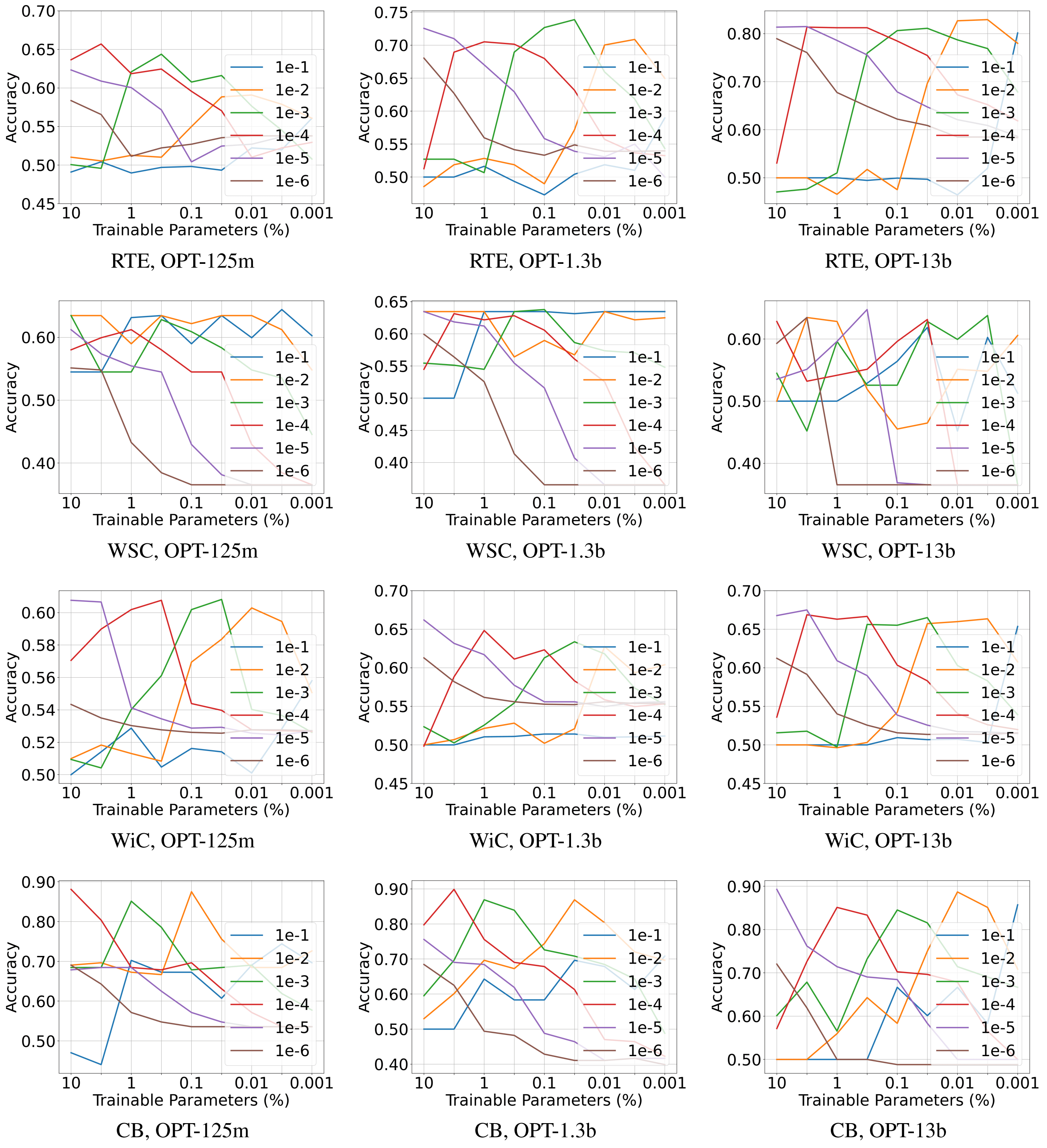
Model	Method	Params	SST-2	RTE	WSC	WiC	CB	BoolQ	MultiRC	COPA	ReCoRD	SQuAD	DROP
OPT-125m	Masking	10%	1e−5	1e−4	1e−2	1e−5	1e−4	1e−5	1e−5	1e−6	1e−6	1e−5	1e−5
		5%	1e−5	1e−4	1e−2	1e−5	1e−4	1e−5	1e−4	1e−6	1e−5	1e−4	1e−4
		1%	1e−4	1e−3	1e−1	1e−4	1e−4	1e−3	1e−4	1e−5	1e−5	1e−4	1e−4
		0.5%	1e−4	1e−3	1e−1	1e−4	1e−3	1e−4	1e−3	1e−5	1e−5	1e−4	1e−3
		0.1%	1e−3	1e−3	1e−2	1e−3	1e−2	1e−3	1e−3	1e−4	1e−4	1e−3	1e−3
		0.05%	1e−3	1e−3	1e−1	1e−3	1e−2	1e−2	1e−3	1e−3	1e−5	1e−3	1e−3
		0.01%	1e−2	1e−2	1e−2	1e−2	1e−1	1e−3	1e−2	1e−3	1e−3	1e−2	1e−2
		0.005%	1e−2	1e−2	1e−1	1e−2	1e−1	1e−2	1e−2	1e−2	1e−4	1e−2	1e−2
		0.001%	1e−1	1e−1	1e−1	1e−1	1e−2	1e−2	1e−1	1e−2	1e−6	1e−1	1e−1
OPT-1.3b	Masking	10%	1e−6	1e−5	1e−2	1e−5	1e−5	1e−5	1e−5	1e−5	1e−6	1e−5	1e−5
		5%	1e−5	1e−5	1e−2	1e−5	1e−4	1e−5	1e−5	1e−5	1e−5	1e−5	1e−5
		1%	1e−5	1e−4	1e−1	1e−4	1e−4	1e−5	1e−4	1e−4	1e−5	1e−4	1e−4
		0.5%	1e−4	1e−4	1e−1	1e−4	1e−3	1e−5	1e−4	1e−3	1e−4	1e−4	1e−3
		0.1%	1e−4	1e−3	1e−3	1e−4	1e−3	1e−4	1e−3	1e−2	1e−4	1e−3	1e−3
		0.05%	1e−3	1e−3	1e−1	1e−3	1e−2	1e−3	1e−3	1e−3	1e−3	1e−3	1e−3
		0.01%	1e−2	1e−2	1e−1	1e−2	1e−2	1e−2	1e−2	1e−2	1e−3	1e−2	1e−2
		0.005%	1e−2	1e−2	1e−1	1e−2	1e−2	1e−2	1e−2	1e−2	1e−3	1e−2	1e−2
		0.001%	1e−2	1e−2	1e−1	1e−2	1e−1	1e−2	1e−2	1e−1	1e−2	1e−2	1e−1
OPT-13b	Masking	10%	1e−5	1e−5	1e−4	1e−5	1e−5	1e−6	1e−5	1e−5	1e−6	1e−5	1e−5
		5%	1e−5	1e−5	1e−2	1e−5	1e−5	1e−5	1e−5	1e−5	1e−5	1e−5	1e−5
		1%	1e−4	1e−4	1e−2	1e−4	1e−4	1e−5	1e−4	1e−4	1e−6	1e−4	1e−4
		0.5%	1e−4	1e−4	1e−5	1e−4	1e−4	1e−4	1e−4	1e−3	1e−5	1e−4	1e−4
		0.1%	1e−3	1e−3	1e−4	1e−3	1e−3	1e−3	1e−3	1e−3	1e−5	1e−3	1e−3
		0.05%	1e−3	1e−3	1e−4	1e−3	1e−3	1e−3	1e−3	1e−3	1e−5	1e−3	1e−3
		0.01%	1e−2	1e−2	1e−3	1e−2	1e−2	1e−2	1e−2	1e−2	1e−4	1e−2	1e−2
		0.005%	1e−2	1e−2	1e−3	1e−2	1e−2	1e−2	1e−2	1e−2	1e−4	1e−2	1e−2
		0.001%	1e−1	1e−1	1e−2	1e−1	1e−1	1e−1	1e−1	1e−1	1e−2	1e−1	1e−1

Table 7: **The optimal learning rate of baselines**, which are obtained via grid search in {1e−1, 1e−2, 1e−3, 1e−4, 1e−5, 1e−6}. Here, FT stands for full parameter fine-tuning, Prefix stands for Prefix-Tuning.

Model	Method	Params	SST-2	RTE	WSC	WiC	CB	BoolQ	MultiRC	COPA	ReCoRD	SQuAD	DROP
OPT-125m	FT	100%	1e−5	1e−5	1e−3	1e−6	1e−5	1e−6	1e−5	1e−6	1e−6	1e−5	1e−5
	Adapter	0.265%	1e−4	1e−4	1e−1	1e−4	1e−5	1e−3	1e−4	1e−5	1e−5	1e−4	1e−4
	LoRA	0.235%	1e−5	1e−4	1e−2	1e−4	1e−3	1e−4	1e−4	1e−4	1e−6	1e−4	1e−3
	AdaLoRA	0.235%	1e−3	1e−3	1e−2	1e−3	1e−3	1e−3	1e−3	1e−4	1e−4	1e−3	1e−3
	Prefix	0.074%	1e−2	1e−3	1e−2	1e−3	1e−2	1e−3	1e−2	1e−2	1e−4	1e−2	1e−4
	BitFit	0.066%	1e−4	1e−4	1e−2	1e−4	1e−5	1e−3	1e−4	1e−5	1e−5	1e−4	1e−4
OPT-1.3b	FT	100%	1e−6	1e−6	1e−6	1e−6	1e−5	1e−6	1e−5	1e−6	1e−6	1e−6	1e−6
	Adapter	0.134%	1e−5	1e−4	1e−2	1e−4	1e−3	1e−4	1e−4	1e−4	1e−5	1e−4	1e−4
	LoRA	0.120%	1e−5	1e−4	1e−3	1e−4	1e−3	1e−4	1e−4	1e−3	1e−5	1e−4	1e−4
	AdaLoRA	0.120%	1e−4	1e−3	1e−2	1e−3	1e−3	1e−3	1e−3	1e−4	1e−4	1e−4	1e−3
	Prefix	0.037%	1e−2	1e−2	1e−2	1e−3	1e−2	1e−2	1e−2	1e−3	1e−4	1e−3	1e−2
	BitFit	0.034%	1e−4	1e−4	1e−2	1e−4	1e−3	1e−4	1e−4	1e−4	1e−5	1e−4	1e−3
OPT-13b	FT	100%	1e−6	1e−5	1e−5	1e−5	1e−5	1e−6	1e−5	1e−6	1e−6	1e−6	1e−6
	Adapter	0.057%	1e−4	1e−3	1e−4	1e−4	1e−3	1e−4	1e−3	1e−4	1e−5	1e−4	1e−4
	LoRA	0.051%	1e−4	1e−3	1e−3	1e−4	1e−4	1e−4	1e−3	1e−4	1e−5	1e−4	1e−4
	AdaLoRA	0.051%	1e−3	1e−3	1e−2	1e−3	1e−3	1e−3	1e−3	1e−3	1e−4	1e−3	1e−3
	Prefix	0.016%	1e−3	1e−3	1e−2	1e−3	1e−2	1e−3	1e−3	1e−3	1e−4	1e−3	1e−3
	BitFit	0.014%	1e−4	1e−3	1e−1	1e−3	1e−3	1e−3	1e−3	1e−4	1e−4	1e−3	1e−3

Table 8: **The complete results of Structured Masking,** with trainable parameter ratio from 0.1 to 0.00001. Here, Masking(S) stands for Structured Masking.

Model	Method	Params	SST-2	RTE	WSC	WiC	CB	BoolQ	MultiRC	COPA	ReCoRD	SQuAD	DROP	Avg
OPT-125m	Masking(S)	10%	87.2	61.0	63.5	60.6	87.5	63.4	63.1	67.0	51.3	64.4	22.9	62.90
		5%	86.7	63.1	63.5	59.8	84.5	63.2	64.3	68.0	51.4	63.8	22.4	62.79
		1%	87.2	63.4	63.5	58.6	80.4	63.1	63.3	67.7	51.3	63.6	22.2	62.21
		0.5%	87.7	63.4	63.5	59.8	76.2	62.6	63.2	67.0	51.2	63.2	23.2	61.91
		0.1%	87.2	59.4	63.5	59.1	75.0	62.6	62.9	68.7	51.3	62.3	21.7	61.24
		0.05%	86.3	58.5	62.5	60.9	73.8	62.5	60.4	68.3	51.4	61.3	20.7	60.60
		0.01%	83.3	57.2	62.2	55.3	67.9	61.2	60.2	65.0	51.4	55.7	20.6	58.16
		0.005%	83.7	57.9	58.3	56.5	67.3	60.2	59.7	66.0	51.4	54.7	20.4	57.83
		0.001%	79.0	53.9	59.6	54.2	67.9	60.4	56.8	67.0	51.2	41.1	16.7	55.25
OPT-1.3b	Masking(S)	10%	93.4	71.5	63.5	61.9	82.7	73.0	68.8	76.7	71.8	82.2	30.3	70.52
		5%	93.5	73.0	63.5	63.9	89.3	71.4	68.6	75.7	72.0	81.9	29.6	71.14
		1%	93.2	69.6	63.5	63.7	83.3	70.3	70.0	76.3	72.5	82.1	29.6	70.38
		0.5%	93.3	72.9	63.1	63.1	82.7	69.8	67.0	76.7	71.8	81.7	30.5	70.24
		0.1%	93.3	69.6	63.5	64.6	71.4	69.0	68.7	75.0	71.2	82.3	29.2	68.87
		0.05%	93.6	72.6	63.5	62.4	81.0	71.4	67.5	76.0	70.8	82.0	29.0	69.97
		0.01%	93.5	71.8	62.8	58.4	69.0	66.6	64.5	74.3	71.4	81.0	30.0	67.59
		0.005%	92.9	69.3	62.2	60.9	63.7	67.7	61.5	75.3	71.4	79.8	28.8	66.66
		0.001%	93.0	68.8	60.9	59.3	69.6	63.7	58.2	73.7	71.4	78.9	27.3	65.90
OPT-13b	Masking(S)	10%	94.8	83.5	63.5	63.5	73.2	76.3	73.8	90.0	81.2	87.9	35.8	74.87
		5%	95.7	81.2	63.5	66.5	78.6	80.4	73.8	89.0	81.9	88.6	34.3	75.76
		1%	94.8	81.2	63.5	62.5	73.2	80.5	74.8	89.0	81.4	87.3	34.3	74.77
		0.5%	95.1	81.1	63.5	67.6	75.0	79.4	73.7	90.0	81.5	88.2	35.3	75.48
		0.1%	94.9	81.1	63.5	63.2	73.2	79.7	74.3	89.0	81.4	88.0	35.7	74.90
		0.05%	94.0	81.6	63.5	66.0	75.0	78.8	69.5	86.0	81.5	88.4	35.1	74.49
		0.01%	95.2	81.8	61.5	62.4	73.2	74.1	72.6	87.0	81.7	88.2	33.3	73.73
		0.005%	93.9	79.3	60.6	66.3	71.4	65.7	71.1	87.0	81.8	87.8	33.5	72.58
		0.001%	93.2	80.7	62.5	61.3	69.6	67.4	62.5	85.0	82.2	87.6	34.2	71.47



RTE, OPT-125m

RTE, OPT-1.3b

RTE, OPT-13b

WSC, OPT-125m

WSC, OPT-1.3b

WSC, OPT-13b

WiC, OPT-125m

WiC, OPT-1.3b

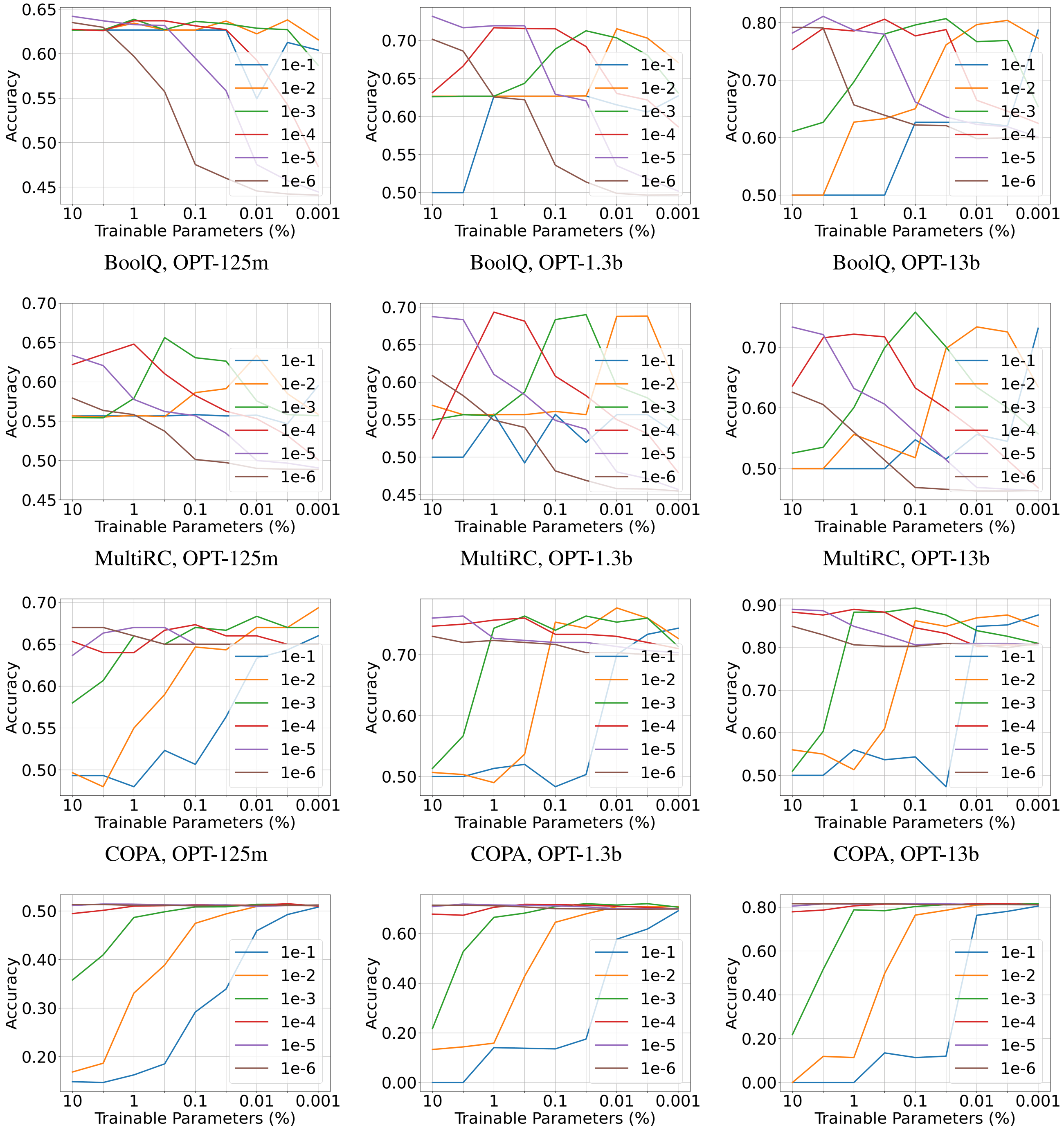
WiC, OPT-13b

CB, OPT-125m

CB, OPT-1.3b

CB, OPT-13b

Figure 6: **The accuracy of Random Masking with different learning rates (part I)**. The x-axis is the percentage of trainable parameters, ranging from {10%, 5%, 1%, 0.5%, 0.1%, 0.05%, 0.01%, 0.005%, 0.001%}. From top to bottom: RTE, WSC, WiC, CB. From left to right: OPT-125m, OPT-1.3b, OPT-13b. The figures show that for Random Masking, smaller trainable parameter ratio requires larger learning rate.



BoolQ, OPT-125m

BoolQ, OPT-1.3b

BoolQ, OPT-13b

MultiRC, OPT-125m

MultiRC, OPT-1.3b

MultiRC, OPT-13b

COPA, OPT-125m

COPA, OPT-1.3b

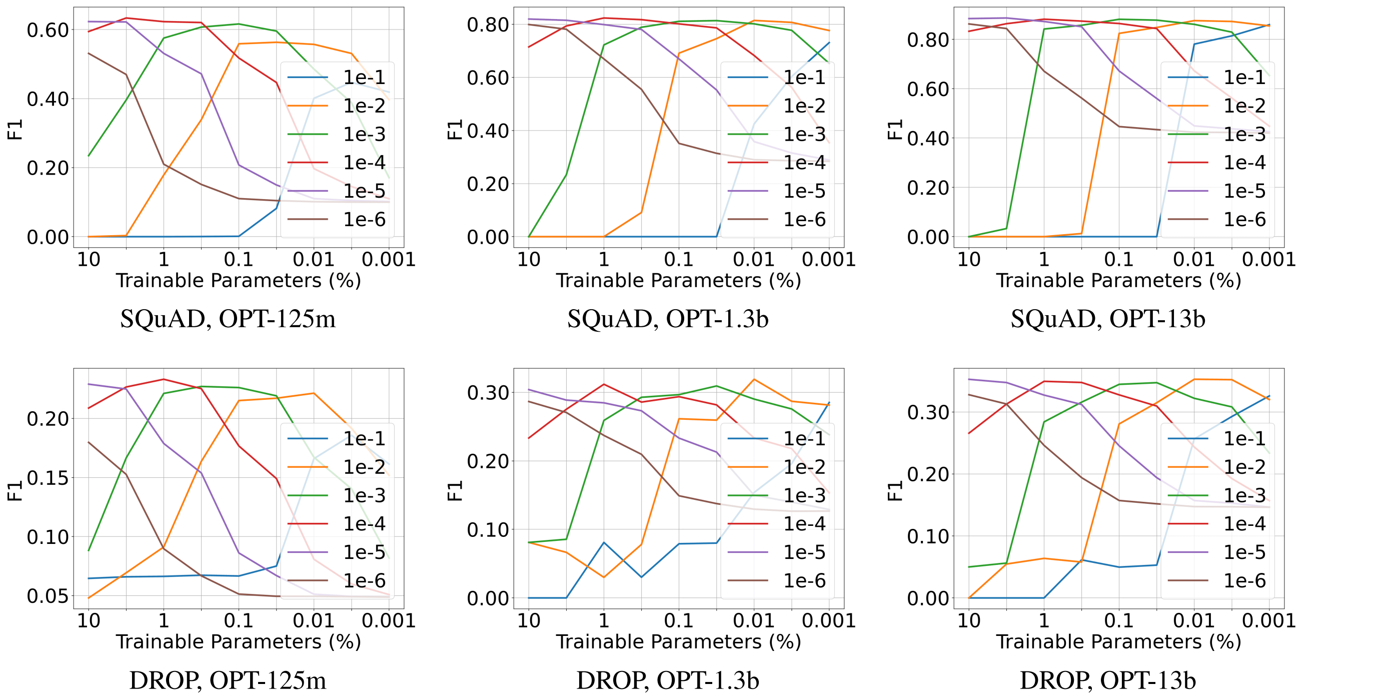
COPA, OPT-13b

ReCoRD, OPT-125m

ReCoRD, OPT-1.3b

ReCoRD, OPT-13b

Figure 7: **The accuracy of Random Masking with different learning rates (part II)**. The x-axis is the percentage of trainable parameters, ranging from $\{10\%, 5\%, 1\%, 0.5\%, 0.1\%, 0.05\%, 0.01\%, 0.005\%, 0.001\%\}$. From top to bottom: BoolQ, MultiRC, Copa, ReCoRD. From left to right: OPT-125m, OPT-1.3b, OPT-13b. The figures show that for Random Masking, smaller trainable parameter ratio requires larger learning rate.



SQuAD, OPT-125m

SQuAD, OPT-1.3b

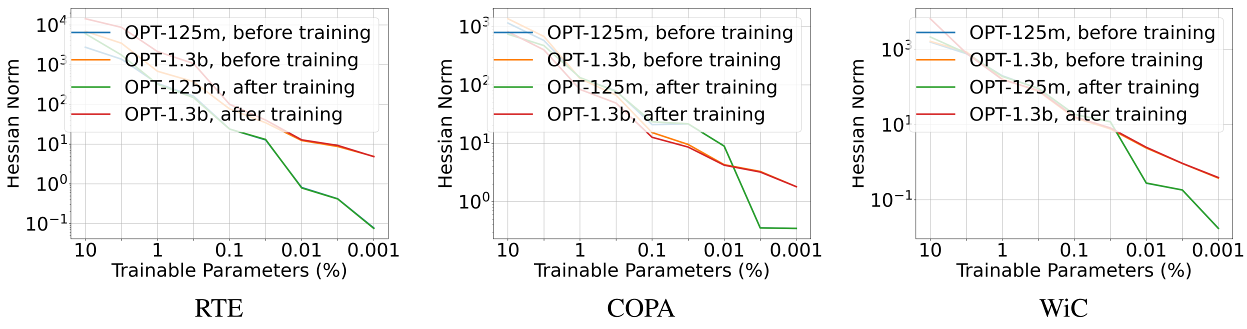
SQuAD, OPT-13b

DROP, OPT-125m

DROP, OPT-1.3b

DROP, OPT-13b

Figure 8: **The accuracy of Random Masking with different learning rates (part III)**. The x-axis is the percentage of trainable parameters, ranging from $\{10\%, 5\%, 1\%, 0.5\%, 0.1\%, 0.05\%, 0.01\%, 0.005\%, 0.001\%\}$. From top to bottom: SQuAD, DROP. From left to right: OPT-125m, OPT-1.3b, OPT-13b. The figures show that for Random Masking, smaller trainable parameter ratio requires larger learning rate.



RTE

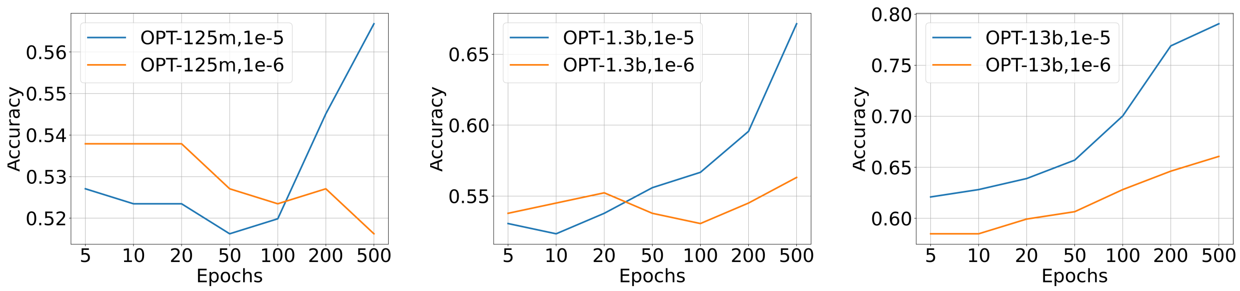
COPA

WiC

Figure 9: **Smaller trainable parameter ratio induces smaller hessian ℓ_2 norm .** These figures are analogs of Figure 4(a) on datasets RTE, COPA, WiC.

Table 9: **The optimal learning rate of Structured Masking**, which are obtained via grid search in $\{1\text{e}{-}1, 1\text{e}{-}2, 1\text{e}{-}3, 1\text{e}{-}4, 1\text{e}{-}5, 1\text{e}{-}6\}$. Here, Masking(S) stands for Structured Masking. This table shows that similar to Random Masking, the optimal learning rate for Structured Masking also has a negative relationship with the number of trainable parameters.

Model	Method	Params	SST-2	RTE	WSC	WiC	CB	BoolQ	MultiRC	COPA	ReCoRD	SQuAD	DROP
OPT-125m	Masking	10%	1e−5	1e−5	1e−2	1e−5	1e−4	1e−3	1e−5	1e−6	1e−6	1e−5	1e−5
		5%	1e−5	1e−4	1e−1	1e−5	1e−4	1e−4	1e−4	1e−5	1e−6	1e−4	1e−4
		1%	1e−4	1e−4	1e−1	1e−4	1e−3	1e−4	1e−4	1e−5	1e−5	1e−4	1e−4
		0.5%	1e−4	1e−4	1e−1	1e−3	1e−3	1e−1	1e−4	1e−5	1e−6	1e−4	1e−3
		0.1%	1e−3	1e−3	1e−1	1e−3	1e−2	1e−1	1e−3	1e−4	1e−4	1e−3	1e−3
		0.05%	1e−3	1e−3	1e−2	1e−3	1e−2	1e−2	1e−3	1e−4	1e−4	1e−3	1e−3
		0.01%	1e−2	1e−3	1e−2	1e−2	1e−2	1e−2	1e−2	1e−3	1e−3	1e−2	1e−2
		0.005%	1e−2	1e−2	1e−2	1e−2	1e−2	1e−2	1e−2	1e−3	1e−2	1e−2	1e−2
		0.001%	1e−2	1e−2	1e−1	1e−2	1e−1	1e−1	1e−2	1e−1	1e−6	1e−1	1e−1
OPT-1.3b	Masking	10%	1e−6	1e−5	1e−3	1e−6	1e−4	1e−5	1e−5	1e−5	1e−6	1e−5	1e−5
		5%	1e−5	1e−5	1e−3	1e−5	1e−4	1e−5	1e−5	1e−4	1e−5	1e−5	1e−5
		1%	1e−4	1e−4	1e−1	1e−4	1e−3	1e−4	1e−4	1e−4	1e−4	1e−4	1e−4
		0.5%	1e−4	1e−4	1e−4	1e−4	1e−3	1e−4	1e−4	1e−4	1e−4	1e−4	1e−4
		0.1%	1e−4	1e−3	1e−3	1e−3	1e−3	1e−3	1e−3	1e−3	1e−3	1e−4	1e−4
		0.05%	1e−4	1e−3	1e−1	1e−3	1e−2	1e−3	1e−3	1e−3	1e−3	1e−4	1e−3
		0.01%	1e−3	1e−3	1e−2	1e−3	1e−3	1e−2	1e−2	1e−3	1e−3	1e−3	1e−3
		0.005%	1e−3	1e−2	1e−2	1e−2	1e−2	1e−2	1e−2	1e−2	1e−3	1e−2	1e−3
		0.001%	1e−2	1e−2	1e−1	1e−2	1e−1	1e−2	1e−2	1e−1	1e−3	1e−2	1e−2
OPT-13b	Masking	10%	1e−5	1e−5	1e−5	1e−5	1e−6	1e−5	1e−5	1e−5	1e−6	1e−6	1e−5
		5%	1e−5	1e−5	1e−4	1e−5	1e−4	1e−5	1e−5	1e−5	1e−6	1e−5	1e−5
		1%	1e−4	1e−4	1e−4	1e−4	1e−5	1e−3	1e−4	1e−4	1e−5	1e−4	1e−4
		0.5%	1e−4	1e−4	1e−3	1e−4	1e−3	1e−4	1e−4	1e−4	1e−6	1e−4	1e−3
		0.1%	1e−3	1e−3	1e−1	1e−3	1e−4	1e−3	1e−3	1e−2	1e−5	1e−3	1e−3
		0.05%	1e−3	1e−3	1e−1	1e−3	1e−2	1e−3	1e−3	1e−2	1e−4	1e−3	1e−3
		0.01%	1e−3	1e−3	1e−3	1e−3	1e−2	1e−3	1e−2	1e−2	1e−4	1e−3	1e−3
		0.005%	1e−2	1e−2	1e−3	1e−2	1e−2	1e−2	1e−2	1e−2	1e−4	1e−2	1e−3
		0.001%	1e−2	1e−2	1e−1	1e−2	1e−2	1e−2	1e−1	1e−2	1e−3	1e−2	1e−2

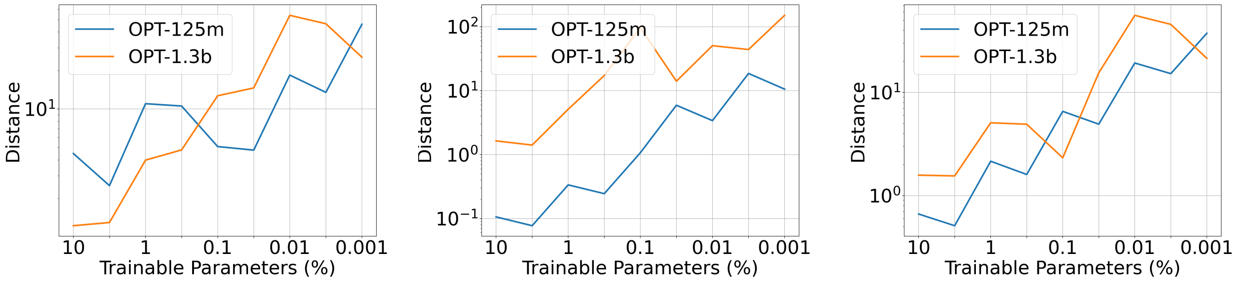


RTE, OPT-125m

RTE, OPT-1.3b

RTE, OPT-13b

Figure 10: **Longer training steps compensate small learning rate.** These figures are analogs of Figure 4(b), with different learning rates and model sizes.



RTE COPA WiC

Figure 11: **Smaller trainable parameter ratio gives more distant solutions.** These figures are analogs of Figure 4(c) on datasets RTE, COPA, WiC.