FINE-TUNING DISTORTS PRETRAINED FEATURES AND UNDERPERFORMS OUT-OF-DISTRIBUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

When transferring a pretrained model to a downstream task, two popular methods are fine-tuning (updating all the model parameters) and linear probing (updating only the last linear layer). It is well known that fine-tuning leads to better accuracy in-distribution (ID). However, in this paper, we show that fine-tuning can achieve worse accuracy than linear probing out-of-distribution (OOD), especially when the pretrained features are good and distribution shift is large. On six distribution shift datasets (Breeds-Living17, Breeds-Entity30, DomainNet, CIFAR $\rightarrow$ STL, CIFAR10.1, FMoW), fine-tuning obtains an average 2% higher accuracy ID but 6% lower accuracy OOD than linear probing. We theoretically analyze the tradeoffs arising in fine-tuning overparameterized two-layer linear networks, characterizing how fine-tuning can distort high-quality pretrained features which leads to low OOD accuracy. Our analysis suggests that the simple two-step strategy of linear probing then full fine-tuning combines the benefits of both fine-tuning and linear probing to achieve better ID and OOD accuracy than fine-tuning, both theoretically and on the above datasets (1% better ID, 8% better OOD).

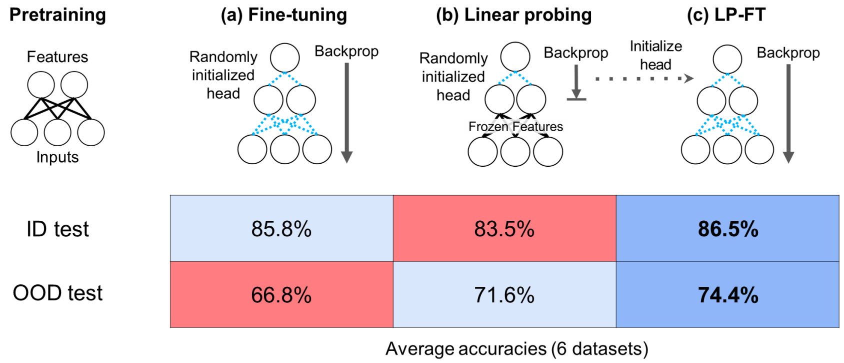
1 INTRODUCTION

Pretraining a model on a large dataset before transferring to a downstream task's training data substantially improves accuracy over training from scratch—for example, pretraining a ResNet-50 on unlabeled ImageNet boosts accuracy on CIFAR-10 from 94% to 98% (Chen et al., 2020a;b). High-stakes applications such as poverty mapping in under-resourced countries (Jean et al., 2016), self-driving cars (Yu et al., 2020), and medical diagnosis (AlBadawy et al., 2018), require models that also generalize to circumstances not seen in the training distribution. In addition to testing on data drawn from the downstream task's training distribution (in-distribution; ID) it is increasingly important to test on data distributions unseen during training (out-of-distribution; OOD).

After initializing with a pretrained model, two popular transfer methods are fine-tuning (running gradient descent on all the model parameters), and linear probing (tuning the head but freezing lower layers). In the ID setting it is well known that fine-tuning leads to better accuracy than linear probing (Kornblith et al., 2019; Zhai et al., 2020; He et al., 2020), and even when testing OOD prior work usually fine-tunes all parameters of their model (Hendrycks et al., 2019a; Miller et al., 2021; Andreassen et al., 2021). Intuitively, fine-tuning all layers of a network can improve pretrained features by tailoring them to the specific task, while linear probing freezes these features.

In this work, we investigate the OOD performance of fine-tuning and linear probing and find that surprisingly, fine-tuning often does *worse* than linear probing in the presence of large distribution shift. We experiment on six distribution shift benchmarks (Breeds Living17, Breeds Entity30, DomainNet, CIFAR $\rightarrow$ STL, CIFAR10.1, FMoW geo-shift), initializing with good pretrained features from MoCo-v2 (Chen et al., 2020b) and CLIP (Radford et al., 2021). While both methods offer gains over training from scratch, fine-tuning improves the average ID accuracy from 84% to 86% but brings down the OOD accuracy from 72% to 67% (Figure 1).

When and why does fine-tuning underperform linear probing? We theoretically consider fine-tuning a two-layer linear network in an overparameterized regression setting where the feature extractor layer has been pretrained to map high-dimensional inputs to useful, lower-dimensional, features. We prove that fine-tuning is worse than linear probing on worst-case OOD inputs when using high quality pretrained features. Even with an infinitesimally small learning rate, fine-tuning distorts pretrained features—the features of ID data are updated while those of OOD data remain unchanged.



	(a) Fine-tuning	(b) Linear probing	(c) LP-FT
ID test	85.8%	83.5%	**86.5%**
OOD test	66.8%	71.6%	**74.4%**

Average accuracies (6 datasets)

Figure 1: Given a good feature extractor (top-left), a randomly initialized head is added to map features to outputs and we can (a) fine-tune all the model parameters or (b) linear-probe, which freezes the feature extractor and trains only the head. We run experiments on six distribution shift datasets. Fine-tuning does well when the test example is sampled from the fine-tuning distribution (ID), but underperforms on test examples sampled from OOD distributions. (c) Our theory says that fine-tuning distorts the pretrained feature extractor leading to poor OOD accuracy, but initializing with a linear probed head fixes this—empirically LP-FT gets better accuracies both ID and OOD.

Since the head and feature extractor are simultaneously optimized during fine-tuning to a configuration that works well on ID training data, the head only accomodates the distorted features of ID points and performs poorly on the unchanged features of OOD points. Interestingly, we show that this feature distortion issue cannot be simply fixed by early stopping—throughout the process of fine-tuning, we never pass through parameters that do well OOD. On the other hand, we show that linear-probing extrapolates better OOD because it preserves pretrained features, but does not do as well as fine-tuning ID because linear probing cannot adapt the features to the downstream task.

Technical challenges. Existing theoretical work on transfer learning focuses on linear probing (Wu et al., 2020; Tripuraneni et al., 2020; Du et al., 2020). In contrast, the analysis of fine-tuning is scarce and challenging because it requires understanding the training dynamics, instead of only the loss function and its global minimizers. In fact, fine-tuning and training from scratch optimize the *same* training loss and only differ in their initializations (pretrained vs random). A mathematical analysis that distinguishes them needs to capture properties of the different global minima that these algorithms converge to, a phenomenon that is sometimes theoretically referred to as the implicit regularization effect of initialization (Neyshabur et al., 2014). Accordingly, our analysis reasons about the parameters that gradient methods pass through starting from the pretrained initialization, which is challenging because there is no known closed form for this trajectory. Two-layer linear networks are widely studied in the implicit regularization community (Saxe et al., 2014; Arora et al., 2018), however they analyze random and often small initializations which don't capture pretraining.

Algorithmic implications. Our theory says that fine-tuning fails because when trying to fit ID training data with a randomly initialized head, the feature extractor changes significantly for ID examples, making features for ID and OOD examples largely inconsistent. This can be fixed by initializing with a good head that does not need to be updated much during fine-tuning, reducing how much the feature extractor changes. This suggests the simple two-step strategy of first linear-probing to find a good head and then full fine-tuning (LP-FT). *Empirically, LP-FT outperforms fine-tuning and linear-probing, both ID and OOD*. Even on CIFAR-10.1 (small distribution shift), where fine-tuning is better for both ID and OOD, we find LP-FT outperforms fine-tuning on both metrics. LP-FT and vanilla fine-tuning use similar amounts of compute because the first step of linear probing is relatively very cheap. We note that while LP-FT has sometimes been used as a fine-tuning heuristic (Kanavati & Tsuneki, 2021), it has not been used for robustness / OOD accuracy, and we show that it addresses the ID-OOD tradeoff theoretically and empirically.

Practical insights. Finally, we check whether fine-tuning fails and LP-FT works, for the reasons predicted by our feature distortion theory. As predicted by the theory, we find that: (1) fine-tuning indeed never matches the OOD accuracy of linear probing throughout the course of training, (2) fine-tuning changes the features for ID examples more than for OOD examples leading to distortions (3) fine-tuning can do better than linear probing OOD if the pretrained features are not very high quality (MoCo-v1 instead of MoCo-v2) or the ID and OOD datasets are very close (e.g., CIFAR-10 and CIFAR-10.1) and (4) LP-FT indeed changes both ID and OOD features orders of magnitude less than fine-tuning does.

2 Setup

Task and evaluation. Given training samples $\{(x_1, y_1), \dots (x_n, y_n)\}$ sampled from some distribution P_{id}, our goal is to learn a predictor $f : \mathbb{R}^d \to \mathcal{Y}$ to map inputs $x \in \mathbb{R}^d$ to outputs $y \in \mathcal{Y}$. We evaluate predictors on their standard "in-distribution" (ID) performance L_{id} on new test samples drawn from P_{id} that the training data is also sampled from. We also evaluate classifiers on their "out-of-distribution" (OOD) performance L_{ood} on test samples drawn from a new distribution P_{ood} that is different from P_{id}. Formally, for some loss function ℓ, we evaluate classifiers on:

$$L_{\mathsf{id}}(f) = \mathop{\mathbb{E}}_{(x,y)\sim P_{\mathsf{id}}}[\ell(f(x), y)] \text{ and } L_{\mathsf{ood}}(f) = \mathop{\mathbb{E}}_{(x,y)\sim P_{\mathsf{ood}}}[\ell(f(x), y)]. \tag{2.1}$$

Models. In this work, we focus on predictors that leverage pretrained feature extractors . For convenience of such analyses, we parameterize the final predictor in terms of a linear "head" $v \in \mathcal{V}$ on top of some features $g_B(x) \in \mathbb{R}^k$ for some "base" parameters $B \in \mathcal{B}$. Formally, f is parameterized by base parameter B and head parameters v such that $f_{v,B}(x) = v^\top g_B(x)$. In our experiments (Section 4), g_B is a deep network and in our theory (Section 3), g_B is a linear projection.

We assume access to some initial pretrained feature extractor B_0 that is obtained by training on potentially large amounts of data from a distribution that could be different from P_{id} and P_{ood}. We focus on two popular methods to learn a predictor $f_{v,B}$ given training data from P_{id}: (i) linear probing where $B = B_0$ and the linear head is obtained by minimizing some loss (e.g., logistic loss for classification, squared loss for regression) on the training data, and (ii) fine-tuning where both v and B are updated by performing gradient descent on some loss on the training data with B initialized at B_0.

3 Theory: fine-tuning distorts pretrained features

We theoretically study the performance of different training methods that use a good pretrained feature extractor. In a linear setting, we characterize when and why fine-tuning, in which all model parameters are updated, can increase OOD loss due to feature distortion. We define the problem setting in Section 3.1, and present the main result and proof sketch in Section 3.2.

Our analysis handles two key challenges which distinguish it from prior work on transfer learning in linear models (Wu et al., 2020; Tripuraneni et al., 2020; Du et al., 2020; Xie et al., 2021a). Prior work focuses on linear probing, while we study fine-tuning where the resulting optimization problem is *non-convex* and we need to analyze the effect of initialization from a particular pretrained parameter setting. We also study *overparameterized models* where the number of training examples is less than the input dimension, reflecting the practical setting where neural networks achieve zero training loss. Here, training loss alone does not determine test performance—this fact makes the setting very relevant because both training from scratch and fine-tuning have same training loss but very different test performance, but also makes the analysis challenging.

3.1 Linear overparameterized setting

Models. Recall from Section 2 that we parameterize predictors in terms of base and head parameters. In this section, we study models where the feature extractor is linear, i.e. $f_{v,B}(x) = v^\top Bx$ where $B \in \mathcal{B} = \mathbb{R}^{k \times d}$, and $v \in \mathcal{V} = \mathbb{R}^k$.

For simplicity, we assume the models are well-specified i.e. $y = v_\star^\top B_\star x$ where $v_\star \in \mathbb{R}^k$ and $B_\star \in \mathbb{R}^{k \times d}$. [1] Note that $B_\star$ and $v_\star$ are only unique up to rotations, i.e., for any rotation matrix U, $(Uv_\star)^T(UB_\star)x = v_\star^T B_\star x$.

[1] We note that our main contribution—analysis of fine-tuning (Theorem 3.1) does not require this well-specified assumption. We compare fine-tuning with linear probing by adapting earlier work on linear probing which requires the well-specified assumption.

Suppose we have a pretrained feature extractor B_0 close to $B_\star$, so $\min_U \|B_0 - UB_\star\|_2 \leq \epsilon$ over rotation matrices $U \in \mathbb{R}^{k \times k}$—this follows from prior work in pretraining (Tripuraneni et al., 2020). As in prior work suppose $B_\star, B_0$ have been orthogonalized to have orthonormal rows.

Data distribution and evaluation. For our analysis, we focus on regression , where $\mathcal{Y} = \mathbb{R}$ and ℓ is the squared loss. Let $X \in \mathbb{R}^{n \times d}, X \neq 0$ be a matrix encoding n training points from $P_{\textsf{id}}$ where each of the n rows is a training input. Let $Y \in \mathbb{R}^n$ be the corresponding outputs. We consider an overparameterized setting where $1 \leq k < n < d$ and $k < d - n$. Intuitively, the input dimension d is high (e.g., 10K), feature dimension k is lower (e.g., 100) and n is in the middle (e.g., 5K).

In this work, we are primarily interested in the out-of-distribution (OOD) performance. Since the OOD data can be arbitrary we follow prior work (Rosenfeld et al., 2021; Kamath et al., 2021; Chen et al., 2021b) and consider the worst case loss over distributions (equivalently, individual points) of bounded norm:

$$L_{\textsf{ood}}(v, B) = \max_{\|x\|_2 \leq 1} (v_\star^\top B_\star x - v^\top B x)^2 = \|B_\star^\top v_\star - B^\top v\|_2^2 \tag{3.1}$$

Training methods. Given training data and a pretrained base parameter B_0, we study the two popular methods of linear probing (LP) and fine-tuning (FT) to learn the final predictor (See Section 2). Both methods involve optimizing the training loss via gradient descent (or variants). In order to effectively analyze these gradient based algorithms, we study vanishing step sizes leading to gradient flows. Gradient flows can be thought of as a continuous time analog of gradient based methods and have been extensively studied in recent years as a way to understand gradient based methods (Gunasekar et al., 2017; Arora et al., 2018; Du et al., 2018). Formally, for training loss $\widehat{L}(v, B) = \|XB^\top v - Y\|_2^2$, the gradient flow differential equations for LP and FT are as follows.

$$\partial_t v_{\textsf{ft}}(t) = -\nabla_v \widehat{L}(v_{\textsf{lp}}(t), B_{\textsf{ft}}(t)),\ \partial_t B_{\textsf{ft}}(t) = -\nabla_B \widehat{L}(v_{\textsf{ft}}(t), B_{\textsf{ft}}(t)) \tag{3.2}$$

$$\partial_t v_{\textsf{lp}}(t) = -\nabla_v \widehat{L}(v_{\textsf{lp}}(t), B_0),\ \partial_t B_{\textsf{lp}}(t) = 0, \tag{3.3}$$

initialized with $B_{\textsf{ft}}(0) = B_{\textsf{lp}}(0) = B_0$ and $v_{\textsf{ft}}(0) = v_{\textsf{lp}}(0) = v_0$. In practice, the head parameter v_0 is initialized randomly—our results hold for any standard random initialization Glorot & Bengio (2010), for example $v_0 \sim \mathcal{N}(0, \sigma^2 I)$ for any σ^2. Recall that the initial value of the base parameter B_0 is assumed to be available and obtained via pretraining.

The final LP and FT solutions are the limit points of the corresponding gradient flows:

$$v_{\textsf{ft}}^\infty = \lim_{t \to \infty} v_{\textsf{ft}}(t) \text{ and } B_{\textsf{ft}}^\infty = \lim_{t \to \infty} B_{\textsf{ft}}(t) \tag{3.4}$$

$$v_{\textsf{lp}}^\infty = \lim_{t \to \infty} v_{\textsf{lp}}(t) \text{ and } B_{\textsf{lp}}^\infty = \lim_{t \to \infty} B_{\textsf{lp}}(t) = B_0 \tag{3.5}$$

3.2 Fine-tuning distorts pretrained features

The more common method of using a pretrained feature extractor is fine-tuning (FT) which typically improves ID performance relative to linear probing (LP). In this section, we show theoretically that FT distorts features leading to poor OOD performance. We first present the key intuitions demonstrating potential issues of FT and then present our formal theorem lower bounding the OOD error of FT .

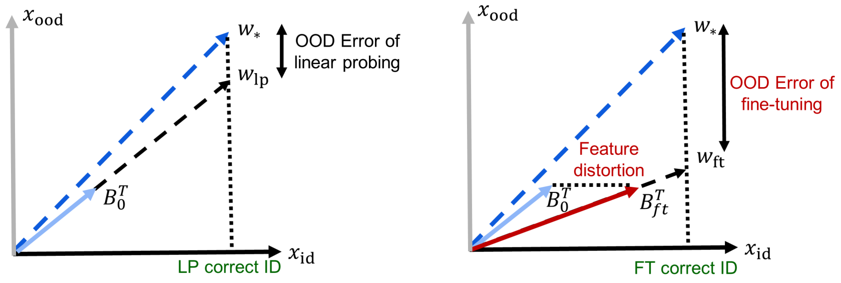
3.2.1 Key intuitions

There are two main pieces that we use to characterize when and why FT has higher OOD error.

1. Features get distorted because representations change only in the ID subspace (i.e., subspace along the training data) and are unchanged in the orthogonal subspace. Taking the derivative of the training loss $\widehat{L}(v, B) = \|XB^\top v - Y\|_2^2$ with respect to the base feature parameter B, we get:

$$\nabla_B \widehat{L}(v, B) = 2v(Y - XBv)^\top X \tag{3.6}$$

By definition, if u is a direction orthogonal to the training subspace, $\nabla_B \widehat{L}(v, B)u = 0$, that is the gradient updates to B do not modify Bu for $u \in S^\perp$. However, the gradient is non-zero for directions u in the ID subspace and the corresponding features Bu change across the FT process. This leads to feature distortion where the features in some subspaces are updated but not others,



(a) Toy example (Linear probing)

(b) Toy example (fine-tuning)

Figure 2: A toy version of our theory illustrating why fine-tuning distorts features, with inputs in 2D. Given input x, the ground truth output is $y = w_\star^\top x$. We have a single training example x_{id} and the pretrained feature extractor is B_0. (a) Linear probing learns w_{lp}, a scaling of the pre-trained features that gets x_{id} correct (w_{lp} and $w_\star$ have the same projection onto x_{id}, vertical dotted line). (b) Fine-tuning updates the pretrained feature extractor along the training direction x_{id} to get B_{ft}, and then learns a scaling of these features that gets x_{id} correct. While both methods get x_{id} correct, fine-tuning makes large errors on x_{ood}, because fine-tuning updates B_0 along x_{id} but not x_{ood}.

leading to inconsistencies relative to the pretrained initialization. Next we discuss the nature and effect of these inconsistencies.

2. Distorted features can lead to higher OOD error. Consider a toy example (Figure 2) where $d = 2$ and the dimensionality of the representations $k = 1$. The linear head is a scalar quantity that denotes how much the features have to be scaled by. Suppose the ID-subspace is the x-axis. There are different ways of fitting the ID subspace depending on the base features as shown in the figure—both fine-tuned and linear probed estimators match the true parameter in the ID subspace. If the base features are optimal or scaled versions of the optimal, constraints on the ID subspace are sufficient to get good performance in all orthogonal subspaces as well. However, in FT, the features change only for inputs in the ID subspace (see (1)) and thus the updated features are *not* simply scaling but distortions where even if the ID error is low, error in subspaces orthogonal to the ID subspace can be high, leading to high worst-case OOD error.

The only way the pretrained features are not distorted and only scaled during FT is if the initial features B_0 is exactly aligned with the ID subspace (i.e. the subspace of training data). In Figure 2, if B_0 is along the x-axis, then updating the features exclusively along the x-axis would simply scale the initial features and not distort them. In this case, the OOD error would not change during FT. However, if the angle is non-zero, the updates would lead to distortions. This motivates our mild non-degeneracy condition where we require the "coverage-angle" to be non-zero.

Definition 3.1 (principal-angle)**.** *Let E and F be matrices with orthonormal columns (orthogonal and unit norm) whose columns span R and $S^\perp$. Recall that $k = dim(R)$. We define $\cos\theta_k(R, S^\perp) = \sigma_k(E^T F)$ which is the k-th largest singular value of $E^\top F$.*

3.2.2 General result on the OOD error of fine-tuning

Our main theorem says that the OOD error of fine-tuning is high.

Theorem 3.1. *In the overparameterized linear setting let $S^\perp = rowspace(X)^\perp$, $R = rowspace(B_0)$, and $v_\star, B_\star$ be the optimal parameters with $w_\star = B_\star v_\star$. If $\cos\theta_k(R, S^\perp) > 0$, then for all t the OOD error of the fine-tuning iterates $(B_{\text{ft}}(t), v_{\text{ft}}(t))$ is lower bounded:*

$$\sqrt{L_{\text{ood}}(v_{\text{ft}}(t), B_{\text{ft}}(t))} \geq \frac{\cos\theta_k(R, S^\perp)}{\sqrt{k}} \frac{\min(\varphi, \varphi^2/\|w_\star\|_2)}{(1 + \|w_\star\|_2)^2} - \epsilon, \tag{3.7}$$

where $\varphi^2 = (v_0^\top v_\star)^2 - (v_\star^\top v_\star)^2$ is defined to be inital head alignment error and $\epsilon = \min_U \|B_0 - UB_\star\|_2^2$ (over rotation matrices U) is the error in the pretrained feature extractor.

Proof sketch. Since the features do not change for examples in $S^\perp$ (perpendicular to the training data), we show that in order to achieve low error on $S^\perp$ the linear head $v_{\text{ft}}(t)$ would have to become the optimal $v_\star$ at some time t. The head initialization v_0 is random and likely to be far from $v_\star$

(measured by the alignment error φ), so the the head would have to change a lot for this. As we see from the fine-tuning gradient flow (3.2), $v_{\mathsf{ft}}(t)$ and $B_{\mathsf{ft}}(t)$ change in a "coupled" manner, and a 'balancedness' invariant in Du et al. (2018) holds across the fine-tuning trajectory. Correspondingly, if $v_{\mathsf{ft}}(t)$ changes a lot the features $B_{\mathsf{ft}}(t)$ also change a lot—we show that this change would lead to high error on other examples (specifically, examples in S). Either ways, fine-tuning would get some subspace of examples wrong, leading to high OOD error. The full proof appears in Appendix A.

Interpretations of various quantities. *Quality of pretrained features (ϵ).* To unpack the bound consider a special case where the pretrained features are perfect ($\epsilon = 0$). With perfect features, Proposition A.2 shows that linear probing gets zero OOD error. Theorem 3.1 shows that $L_{\mathsf{ood}}(v_{\mathsf{ft}}(t), B_{\mathsf{ft}}(t)) > 0$ at all times t—so fine-tuning underperforms when the features are perfect.

Alignment error of random head initialization (φ). The lower bound increases as φ increases i.e. alignment error increases. The gradient updates to the head and base parameters are coupled. If the head was initialized perfectly at $v_\star$, then fine-tuning updates would not increase the OOD error. However, when the head is randomly initialized as is standard in fine-tuning, the alignment error is high, leading to high OOD error. We use this insight in Section 3.4 to show that smarter head initialization (namely via first linear probing) improves OOD performance of fine-tuning.

3.3 Linear probing vs fine-tuning

In this section, we use our main theorem on fine-tuning (Theorem 3.1) and adapt prior work on linear probing to show for a simple Gaussian data distribution that LP is better than FT, OOD. We assume each training example $X_i \sim \mathcal{N}(0, I)$.

Theorem 3.2. *In the linear overparameterized setting, suppose the training data is Gaussian. and recall that ϵ is the error in the pretrained feature extractor. Then as the feature extractor error ϵ goes to 0, linear probing does much better than fine-tuning OOD:*

$$\frac{L_{\mathsf{ood}}(v_{\mathsf{lp}}^\infty, B_0)}{L_{\mathsf{ood}}(v_{\mathsf{ft}}(t), B_{\mathsf{ft}}(t))} \xrightarrow{p} 0, as \;\; \epsilon \to 0 \tag{3.8}$$

This holds for all times t for FT (and therefore also for the limit $v_{\mathsf{ft}}^\infty, B_{\mathsf{ft}}^\infty$) and the LP iterates converge to $v_{\mathsf{lp}}^\infty, B_0$ as a result of the gradient flow on a convex problem.

Intuitively, if the pretrained features are good, LP learns the optimal linear head which has small OOD error while Theorem 3.1 provides a lower bound on the OOD error for fine-tuning. In Appendix A, we also give a threshold T (in terms of d, n, k) where LP does better than FT if $\epsilon < T$.

ID vs OOD error tradeoffs. Until now, we focused on the OOD error and showed how FT can have higher OOD error than LP. However, in practice, we also care about the in-distribution ID error. How do the two methods compare in their ID performance?

For simplicity, we consider the "ID subspace loss" which measures the maximum loss over points in the subspace of training data (S). This allows us to work without distributional assumptions on P_{id} but it is straightforward to extend to particular distributions. See Appendix A for a formal definition and relationship between ID subspace loss and ID test loss for Gaussian distribution.

If the pretrained initialization is perfect, i.e. $B_0 = B_\star$, then both LP and FT get zero $L_{\mathsf{id:subspace}}$ as they can fit the training data perfectly in an overparameterized setting. But if $B_0 \neq B_\star$, there may not be a linear head on B_0 that fits the training data perfectly and LP can have high ID error. FT, on the other hand, can update the features to find a new B_{ft}^∞ that can fit the training data perfectly with a linear head v_{ft}^∞. We state the formal proposition below relating the ID errors of the two methods.

Proposition 3.1. *Suppose $w_\star = B_\star^\top v_\star \notin$ rowspace(B_0), and that fine-tuning converges to a local minimum of its loss, then fine-tuning does better ID: $L_{\mathsf{id:subspace}}(v_{\mathsf{ft}}^\infty, B_{\mathsf{ft}}^\infty) < L_{\mathsf{id:subspace}}(v_{\mathsf{lp}}^\infty, B_0)$.*

To summarize, we proved that in a simple Gaussian setting, there are tradeoffs between ID and OOD error: FT has lower ID error but higher OOD error than LP. In the next section, we extend our theoretical insights to show that a simple variant of FT can mitigate such tradeoffs.

3.4 Linear probing then fine-tuning: a simple variant to mitigate tradeoffs

The advantage of fine-tuning is it can adapt both the feature extractor and head to fit the downstream task. Can we keep this benefit while ensuring that our OOD error is low when we have good pretrained features?

Going back to Theorem 3.1, we see that the alignment error in the head initialization $\varphi^2 = (v_0^\top v_\star)^2 - (v_\star^\top v_\star)^2$ plays an important role. The issue with FT was that under random initialization, φ is usually large and since the gradient updates to the base parameter are coupled with that of the head parameter, the base features get distorted in a manner that increases the OOD error. This suggests that we should use a better head initialization—one obtained from linear probing. If the pretrained features are decent, a linear probed head would be much better aligned with $v_\star$ allowing the base features to be updated in a manner that does not increase the OOD error. We formally prove this intuition in a simple setting below.

Proposition 3.2. *Suppose we have perfect pretrained features $B_0 = UB_\star$ for some rotation U. Let $R = \mathit{rowspace}(B_0)$. Under the non-degeneracy conditions $\cos\theta_k(R, S) \neq 0, \cos\theta_k(R, S^\perp) \neq 0$:*

$$\forall t, L_{\mathsf{ood}}(B_{\mathsf{ft}}(t)^\top v_{\mathsf{ft}}(t)) > 0, \text{ if } v_0 \sim \mathcal{N}(0, \sigma^2 I) \text{ is randomly initialized (FT)} \tag{3.9}$$

$$\forall t, L_{\mathsf{ood}}(B_{\mathsf{ft}}(t)^\top v_{\mathsf{ft}}(t)) = 0, \text{ if } v_0 \text{ is initialized to } v_{\mathsf{lp}}^\infty \text{ (LP-FT)} \tag{3.10}$$

4 Experiments

We run experiments on six benchmark datasets with deep neural networks and see that given good pretrained features, fine-tuning does better ID but worse OOD than linear probing. As predicted by the theory, we find that LP-FT does better than both methods. Finally, we see that a number of predictions from the feature distortion theory hold up in practice. The datasets we use are:

- **DomainNet** (Peng et al., 2019) is a standard domain adaptation dataset. Here, our ID dataset contains 'sketch' images (e.g., drawings of apples, elephants, etc), and the OOD dataset contains 'real', 'clipart', and 'painting' images of the same categories. We use the version of the dataset from Tan et al. (2020).
- **Living-17** and **Entity-30** are sub-population shift datasets from the BREEDS benchmark (Santurkar et al., 2020). For example, in Living-17 the goal is to classify an image as one of 17 animal categories such as 'bear'—the ID dataset contains images of black bears and sloth bears and the OOD dataset has images of brown bears and polar bears.
- **FMoW Geo-shift** is adapted from the satellite remote sensing dataset 'Functional Map of the World' (Christie et al., 2018; Koh et al., 2021). The goal is to classify a satellite image into one of 62 categories such as 'impoverished settlement'. Our ID dataset contains images from North America, and the OOD dataset contains images from Africa and Europe.
- **CIFAR-10 → STL** is another standard domain adaptation dataset (French et al., 2018), where the ID is CIFAR-10 (Krizhevsky, 2009), and the OOD is STL (Coates et al., 2011).
- **CIFAR-10 → CIFAR-10.1** (Recht et al., 2018) is a dataset collected using a very similar protocol to CIFAR-10, and the authors describe it as "a minute distributional shift". The hope is that a classifier trained on CIFAR-10 gets high accuracy on CIFAR-10.1.

Pretraining and models. We use a ResNet-50 architecture for our experiments. We consider a diverse range of pretraining methods and datasets: MoCo-v2 (Chen et al., 2020b), CLIP (Radford et al., 2021), and MoCo-TP (Ayush et al., 2020)—see Appendix B for details.

In Appendix B, we also show results for a larger vision transformer architecture, more fine-tuning baselines, and larger scale datasets (linear probing or fine-tuning on ImageNet (Russakovsky et al., 2015) and evaluating on ImageNetV2 (Recht et al., 2019), ImageNet-R (Hendrycks et al., 2020), ImageNet-Sketch (Wang et al., 2019), ImageNet-A (Hendrycks et al., 2019b)).

4.1 Linear probing vs fine-tuning

Experiment protocols. We initialize with the pretrained model, and fine-tune or linear probe on ID training examples. For fine-tuning on each dataset we swept over 6 learning rates, using a cosine learning rate schedule and batch size of 64. We early stop and choose the best learning rate using ID validation accuracy. For linear probing we train an $\ell2$-regularized logistic regression classifier on frozen features from the penultimate layer of the pretrained model, selecting the best $\ell2$-regularization hyperparameter based on ID validation accuracy. For all methods, we run each hyperparameter configuration 3 times (with different random seeds), and take the average accuracy. OOD data was only used for evaluation. For more details, see Appendix B.

Table 1: **ID accuracies** with 90% confidence intervals over 3 runs—fine-tuning does better than linear probing on all datasets except DomainNet. LP-FT does the best on all except FMoW where it is in between linear probing and fine-tuning.

	CIFAR-10	Ent-30	Liv-17	DomainNet	FMoW	Ave
FT	**97.3 (0.2)**	**93.6 (0.2)**	97.1 (0.2)	84.5 (0.6)	**56.5 (0.3)**	85.8
LP	91.8 (0.0)	90.6 (0.2)	96.5 (0.2)	89.4 (0.1)	49.1 (0.0)	83.5
LP-FT	**97.5 (0.1)**	**93.7 (0.1)**	**97.8 (0.2)**	**91.6 (0.0)**	51.8 (0.2)	**86.5**

Table 2: **OOD accuracies** with 90% confidence intervals over 3 runs. Linear probing does better than fine-tuning on all datasets except CIFAR-10.1, where the ID and OOD are very similar. LP-FT matches or exceeds fine-tuning and linear probing on all six OOD datasets.

	STL	CIFAR-10.1	Ent-30	Liv-17	DomainNet	FMoW	Ave
FT	82.4 (0.4)	92.3 (0.4)	60.7 (0.2)	77.8 (0.7)	55.5 (2.2)	32.0 (3.5)	66.8
LP	85.1 (0.2)	82.7 (0.2)	**63.2 (1.3)**	82.2 (0.2)	79.7 (0.6)	**36.6 (0.0)**	71.6
LP-FT	**90.7 (0.3)**	**93.5 (0.1)**	**62.3 (0.9)**	**82.6 (0.3)**	**80.7 (0.9)**	**36.8 (1.3)**	**74.4**

Results. Fine-tuning does better than linear probing on 4 out of 5 ID datasets (average accuracy of 85.8% vs 83.5% for linear probing, see Table 1). This is consistent with prior work and intuitions. However, OOD, linear-probing does better on 5 out of 6 distribution shift datasets (average accuracy of 71.6% for linear probing vs 66.8% for fine-tuning, see Table 2). Our datasets vary in size from 20K examples to 150K examples, so this doesn't appear to be simply because of sample size.

4.2 Linear probing then fine-tuning (LP-FT)

Experiment protocols. For LP-FT, we initialize the neural network head using the linear probed solution, and then fine-tune the model. LP-FT and fine-tuning use similar compute because the linear probing step is much faster than fine-tuning. As with fine-tuning, we sweep over 6 learning rates, early stopping using ID validation accuracy (more details are in Appendix B).

Results. We find that LP-FT gets the best accuracy ID (average: 86.5%) and OOD (average: 74.4%). This is true for 4/5 ID and 6/6 OOD datasets—every dataset except FMoW ID, where LP-FT is better than linear probing but worse than fine-tuning. Since the ID accuracy on FMoW is low (56.5%), this could be because the pretrained features are not good.

4.3 Examining the feature distortion theory

Early stopping. Our theory predicts that fine-tuning would do worse OOD throughout the process of fine-tuning, and not just at the end. Here, we early stop each fine-tuning method and choose the best learning rate based on target validation accuracy. As expected, fine-tuning does improve a little, but linear probing (average accuracy: 73.0%) is still better than fine-tuning (average accuracy: 70.1%). See Appendix B for per-dataset results.

ID-OOD Features get distorted. The feature distortion theory predicts that fine-tuning changes features for ID examples more than for OOD examples, which is why fitting a head on ID examples performs poorly OOD. The theory also predicts that LP-FT changes the features less than than fine-tuning does. For each example in Living-17, we took the Euclidean distance of the ResNet-50 features before and after fine-tuning. As expected, the average distance for ID examples (0.019) is more than for OOD examples (0.017), and both distances are 20× smaller for LP-FT. We show results for all the other datasets in Appendix B.

Pretrained features must be good, ID-OOD far apart. Our theory says that linear probing does better than fine-tuning OOD, but only if the OOD and ID data are quite different, and the pretrained features are good—otherwise fine-tuning can do better OOD by adjusting the feature extractor ID.

Feature quality: We use a checkpoint of MoCo that got 10% worse accuracy (on ImageNet) and compare linear probing and fine-tuning on Living-17. With worse features, both methods do worse, but fine-tuning (96% ID, 71% OOD) does better than linear probing (92% ID and 66% OOD).

ID $\approx$ OOD: We fine-tune / linear probe on CIFAR-10, and test on CIFAR-10.1, a dataset collected using a similar protocol to CIFAR-10. As expected, fine-tuning (92.3%) outperforms linear probing OOD (82.7%). Even in this case LP-FT does the best (93.5%).

5 Related work and discussion

Fine-tuning vs linear probing. Fine-tuning (FT) and linear probing (LP) are popular transfer learning algorithms. There is substantial evidence of FT outperforming LP in-distribution (ID) including recent large-scale investigations (Kornblith et al., 2019; Chen et al., 2021a; Zhai et al., 2020; Chen et al., 2020b) (the only notable exception is in Peters et al. (2019) where LP performs better than FT when using ELMo representations, but worse using BERT). FT is therefore the method of choice for improving accuracy, while LP is used to analyze properties of representations (Peters et al., 2018; Belinkov et al., 2017; Hewitt & Manning, 2019). In our work, we find that FT can underperform LP especially when using high quality pretrained features in the presence of a large distribution shift. There are a variety of other fine-tuning heuristics (Ge & Yu, 2017; Guo et al., 2019; Zhang et al., 2020; Zhu et al., 2020; Jiang et al., 2021; Aghajanyan et al., 2021)—combining our insights with these ideas might lead to better methods.

The benefit of preserving pretrained features. Our work adds to growing evidence that *lightweight* fine-tuning where only small parts of a pretrained model are updated, extrapolates better under distribution shifts. Zero-shot language prompting in vision (Radford et al., 2021) and other lightweight fine-tuning approaches in NLP (Houlsby et al., 2019; Li & Liang, 2021; Xie et al., 2021b; Lester et al., 2021; Utama et al., 2021; Zhou et al., 2021) improve OOD performance. In independent and concurrent work, Andreassen et al. (2021) observe that through the course of fine-tuning, ID accuracy continues to increase but OOD accuracy plateaus.

Mitigating ID-OOD tradeoffs. While LP-FT has sometimes been used as a fine-tuning heuristic (Kanavati & Tsuneki, 2021; fastai), it has not been used for robustness / OOD accuracy, and we show that it addresses the ID-OOD tradeoff theoretically and empirically. Tradeoffs between ID and OOD accuracy are widely studied and prior work self-trains on large amounts of unlabeled data to mitigate such tradeoffs (Raghunathan et al., 2020; Xie et al., 2021a; Khani & Liang, 2021). In contrast, LP-FT uses no extra unlabeled data and is a simple variant of fine-tuning. In concurrent and independent work, Wortsman et al. (2021) show that ensembling the weights of a zero-shot and fine-tuned model mitigates the ID-OOD tradeoff between these approaches, and this method could be promising for our datasets as well.

Theoretical analysis of transfer learning. Prior works look at ID error (Wu et al., 2020; Tripuraneni et al., 2020; Du et al., 2020), while we look at OOD error. In recent work (Chua et al., 2021) study regularized fine-tuning in an underparameterized regime where there is a unique global optimum. In contrast, our analysis deals with the overparameterized regime (mirroring modern settings of zero train loss) where we need to analyze the trajectory of fine-tuning from the pretrained initialization because there is no unique optimizer of the objective function. See Section C for additional related work on theory of overparameterized models.

Conclusion. There is a strong trend towards leveraging pretrained models to improve downstream performance, and whenever feasible, it is common to fine-tune all model parameters. In this work, we show theoretically and empirically that preserving features might be important for robustness, and simpler approaches like linear-probing can improve OOD performance. *This OOD gap between FT and LP grows as the quality of pretrained features improve, so we believe our results are likely to gain significance over time with growing innovations and scale of pretraining.*

Theoretical understanding of modern deep learning remains limited, especially the effect of pretraining and transfer learning—our work introduces some ideas for dealing with these challenges. There are several open questions and extensions such as dealing with non-linear activations, different layerwise learning rates, and the effect of explicit regularization.

Finally, we showed LP-FT can mitigate tradeoffs between ID and OOD accuracy in our context. LP-FT could be useful in other situations, for example in CLIP we could initialize the final layer with the zero-shot classifier and then fine-tune the entire model, as done in concurrent work (Wortsman et al., 2021). LP-FT is just a first step in leveraging the intuition from our theoretical analysis and we hope that this work inspires new methods of leveraging powerful pretrained models.

Reproducibility: We include proofs for our theoretical results in Appendix A, additional experiment details in Appendix B, and include anonymized source code.

REFERENCES

Pierre Antoine Absil, Alan Edelman, and Plamen Koev. On the largest principal angle between random subspaces. *Linear Algebra and its Applications*, 414(1):288–294, 2006.

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations (ICLR)*, 2021.

EA AlBadawy, A Saha, and MA Mazurowski. Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Med Phys.*, 45, 2018.

Anders Andreassen, Yasaman Bahri, Behnam Neyshabur, and Rebecca Roelofs. The evolution of out-of-distribution robustness throughout fine-tuning. *arXiv*, 2021.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning (ICML)*, pp. 244–253, 2018.

Kumar Ayush, Burak Uzkent, Chenlin Meng, Kumar Tanmay, M. Burke, D. Lobell, and Stefano Ermon. Geography-aware self-supervised learning. *arXiv*, 2020.

Peter L. Bartlett, Philip M. Long, G ́abor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *arXiv*, 2019.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. What do neural machine translation models learn about morphology? In *Association for Computational Linguistics (ACL)*, pp. 861–872, 2017.

Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *arXiv*, 2019.

Koby Bibas, Yaniv Fogel, and Meir Feder. A new look at an old problem: A universal learning approach to linear regression. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 2304–2308, 2019.

Tianle Cai, Ruiqi Gao, J. Lee, and Qi Lei. A theory of label propagation for subpopulation shift. In *International Conference on Machine Learning (ICML)*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pp. 1597–1607, 2020a.

Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv*, 2020b.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021a.

Yining Chen, Elan Rosenfeld, Mark Sellke, Tengyu Ma, and Andrej Risteski. Iterative feature matching: Toward provable domain generalization with logarithmic environments. *arXiv*, 2021b.

Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

Kurtland Chua, Qi Lei, and Jason D Lee. How fine-tuning allows for effective meta-learning. *arXiv preprint arXiv:2105.02221*, 2021.

Adam Coates, Andrew Ng, and Honlak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pp. 215–223, 2011.

Simon S. Du, Wei Hu, Sham M. Kakade, Jason D. Lee, and Qi Lei. Few-shot learning via learning the representation, provably. *arXiv*, 2020.

Simon Shaolei Du, Wei Hu, and Jason Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

fastai. fastai tutorial on transfer learning. `https://github.com/fastai/course-v3/blob/master/nbs/dl1/lesson1-pets.ipynb`.

Geoff French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, 2018.

Weifeng Ge and Yizhou Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010.

Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6151–6159, 2017.

Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: Transfer learning through adaptive fine-tuning. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning (ICML)*, 2019a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019b.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Association for Computational Linguistics (ACL)*, 2019.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. *arXiv*, 2019.

Neal Jean, Marshall Burke, Michael Xie, W. Matthew Davis, David B. Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353, 2016.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *International Conference on Learning Representations (ICLR)*, 2021.

Pritish Kamath, Akilesh Tangella, Danica J. Sutherland, and Nathan Srebro. Does invariant risk minimization capture invariance? In *Artificial Intelligence and Statistics (AISTATS)*, 2021.

Fahdi Kanavati and Masayuki Tsuneki. Partial transfusion: on the expressive influence of trainable batch norm parameters for transfer learning. In *Medical Imaging with Deep Learning*, 2021.

Fereshte Khani and Percy Liang. Removing spurious features can hurt accuracy and affect groups disproportionately. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2021.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021.

Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Thomas Laurent and James H. von Brecht. Deep linear neural networks with arbitrary loss: All local minima are global. In *International Conference on Machine Learning (ICML)*, 2018.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Association for Computational Linguistics (ACL)*, 2021.

Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning (ICML)*, 2018.

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.

John Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning (ICML)*, 2021.

Vidya Muthukumar, Kailas Vodrahalli, Vignesh Subramanian, and Anant Sahai. Harmless interpolation of noisy data in regression. *IEEE Journal on Selected Areas in Information Theory*, 1(1): 67–83, 2020.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv*, 2014.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *International Conference on Computer Vision (ICCV)*, 2019.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *North American Association for Computational Linguistics (NAACL)*, 2018.

Matthew E Peters, Sebastian Ruder, and Noah A Smith. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pp. 7–14, 2019.

Viraj Prabhu, Shivam Khare, Deeksha Karthik, and Judy Hoffman. Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *International Conference on Computer Vision (ICCV)*, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, volume 139, pp. 8748–8763, 2021.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C. Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2020.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do CIFAR-10 classifiers generalize to CIFAR-10? *arXiv*, 2018.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, 2019.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. *arXiv*, 2020.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv*, 2014.

Shuhan Tan, Xingchao Peng, and Kate Saenko. Class-imbalanced domain adaptation: An empirical odyssey. *arXiv preprint arXiv:1910.10320*, 2020.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *arXiv preprint arXiv:2007.00644*, 2020.

Nilesh Tripuraneni, Michael I. Jordan, and Chi Jin. On the theory of transfer learning: The importance of task diversity. *arXiv*, 2020.

Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8:1–230, 2015.

Prasetya Ajie Utama, Nafise Sadat Moosavi, Victor Sanh, and Iryna Gurevych. Avoiding inference heuristics in few-shot prompt-based finetuning. *arXiv preprint arXiv:2109.04144*, 2021.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. *arXiv preprint arXiv:2109.01903*, 2021.

Sen Wu, Hongyang R. Zhang, and Christopher Ré. Understanding and improving information transfer in multi-task learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Sang Michael Xie, Ananya Kumar, Robert Jones, Fereshte Khani, Tengyu Ma, and Percy Liang. In-N-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. In *International Conference on Learning Representations (ICLR)*, 2021a.

Sang Michael Xie, Tengyu Ma, and Percy Liang. Composed fine-tuning: Freezing pre-trained denoising autoencoders for improved generalization. In *International Conference on Machine Learning (ICML)*, 2021b.

Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv*, 2020.

Jeffrey O Zhang, Alexander Sax, Amir Zamir, Leonidas Guibas, and Jitendra Malik. Side-tuning: A baseline for network adaptation via additive side networks. In *European Conference on Computer Vision (ECCV)*, 2020.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*, 2021.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. FreeLB: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations (ICLR)*, 2020.

A Proofs for Section 3

A.1 ID subspace loss

Formally, we define the ID subspace loss as the following:

$$L_{\mathsf{id:subspace}}(v, B) = \max_{\|x\|_2 \leq 1, x \in S} ({v_\star}^\top B_\star x - v^\top Bx)^2 = ||\Pi_S(B_\star^\top v_\star) - \Pi_S(B^\top v)||_2^2, \tag{A.1}$$

where Π_S projects points onto the subspace S, and we consider norm bounded inputs because the linear regression error scales with the norm.

Consider the simple setting of Gaussian distributions where P_{id} is drawn from identity covariance in a low dimensional subspace. In this case, the ID subspace loss is equal to the ID test loss with high probability as the training data would span the entire subspace.

A.2 Feature distortion theorem

We first prove our core theorem, that fine-tuning distorts pretrained features.

Restatement of Theorem 3.1. *In the overparameterized linear setting let $S^\perp = rowspace(X)^\perp$, $R = rowspace(B_0)$, and $v_\star, B_\star$ be the optimal parameters with $w_\star = B_\star v_\star$. If $\cos\theta_k(R, S^\perp) > 0$, then for all t the OOD error of the fine-tuning iterates $(B_{\mathsf{ft}}(t), v_{\mathsf{ft}}(t))$ is lower bounded:*

$$\sqrt{L_{\mathsf{ood}}(v_{\mathsf{ft}}(t), B_{\mathsf{ft}}(t))} \geq \frac{\cos\theta_k(R, S^\perp)}{\sqrt{k}} \frac{\min(\varphi, \varphi^2/\|w_\star\|_2)}{(1 + \|w_\star\|_2)^2} - \epsilon, \tag{A.2}$$

where $\varphi^2 = (v_0^\top v_\star)^2 - (v_\star^\top v_\star)^2$ is defined to be inital head alignment error and $\epsilon = \min_U \|B_0 - UB_\star\|_2^2$ (over rotation matrices U) is the error in the pretrained feature extractor.

We follow the sketch in the main paper. We begin with a few lemmas, showing that certain quantities are preserved throughout the fine-tuning process.

Our first lemma says that the representations $B^t_{ft}x$ do not change for examples perpendicular the span of the training examples. Note that the final output ${v^t_{ft}}^\top B^t_{ft}x$ still changes, because v^t_{ft} changes.

Lemma A.1. *For all times t and all $x \in S^\perp$, we have:*

$$B_0 x = B^t_{ft}x \tag{A.3}$$

Proof. We initialized fine-tuning with the feature extractor $B_{\mathsf{ft}}(0) = B_0$. It suffices to show that $\partial_t B^t_{ft}x = 0$ for all $x \in S^\perp$. Recall that $\partial_t B^t_{ft}$ is given by the gradient flow update equation:

$$\partial_t B^t_{ft} = -\partial_B \widehat{L}(v^t_{ft}, B^t_{ft}) = -\partial_B \|XB^\top v - Y\|_2^2 \tag{A.4}$$

Computing the RHS explicitly using multivariable chain rule, we get:

$$\partial_t B^t_{ft} = -2v(XB^\top v - Y)^\top X \tag{A.5}$$

Since x is a constant, we get:

$$\partial_t B^t_{ft}x = -2v(XB^\top v - Y)^\top Xx \tag{A.6}$$

But $Xx = 0$ for $x \in S^\perp$, since $x \in S^\perp$ is defined as x is perpendicular to the rowspace of X (i.e., perpendicular to the rows of X). So the RHS is 0—that is, $\partial_t B^t_{ft}x = 0$, as desired. □

Next, we show that the change in the head and feature extractor are 'coupled'. So if the head changes in a certain way, then the feature extractor cannot just stay the same. In the literature, this is sometimes called the "balancedness" lemma, and has been proved in prior work on two layer linear networks.

Lemma A.2. *For all t we have:*

$$v_0 v_0^\top - B_0 B_0^\top = v^t_{ft}{v^t_{ft}}^\top - B^t_{ft}{B^t_{ft}}^\top \tag{A.7}$$

Proof. This follows by showing that the derivative is 0:

$$\partial_t[v^t_{ft}{v^t_{ft}}^\top - B^t_{ft}{B^t_{ft}}^\top] = 0 \tag{A.8}$$

Which can be verified by direct calculation. See Theorem 2.2 in Du et al. (2018) and the proof of Theorem 1 in Arora et al. (2018). □

For our proof we will require that every feature $r \in R$ can be generated from some OOD direction, that is $r = B_0 u$ for some $u \in S^\perp$. We will show that this is implied by the condition on the principal angle: $\cos\theta_k(R, S^\perp) > 0$ where $R = \text{rowspace}(B_0)$, which we assumed in Theorem 3.1. The following lemma shows this (and also quantifies that the norm of u does not shrink too much when projected onto R).

Lemma A.3. *Suppose R is a dimension k subspace of $\mathbb{R}^d$ and $S^\perp$ has dimension $d' \geq 1$. Consider any vector $r \in R$ with $\|v\|_2 = \cos\theta_k(R, S^\perp)$. Then there exists $u \in S^\perp$ with $\|u\|_2 \leq 1$ such that $\Pi_R(u) = r$.*

Proof. Let $c = \cos\theta_k(R, S^\perp)$. Firt, we get rid of an easy case—if $c = 0$, then we have $\|r\|_2 = 0$, so $r = 0$. Then we can just pick $u = 0$, and $\Pi_R(u) = 0 = r$. So for the rest of the proof we assume $c > 0$.

Consider arbitrary vector $r \in R$ with $\|r\|_2 \leq c$. Let $E \in \mathbb{R}^{d \times d'}, F \in \mathbb{R}^{d \times k}$ have orthonormal columns, which form a basis for $S^\perp$ and R respectively. We can write $r = Fz$ for some $z \in \mathbb{R}^k$ since the columns of F form a basis for R.

By definition of $\cos\theta_k$ (Definition 3.1), we have that $c = \sigma_k(F^\top E)$. So $\sigma_k(F^\top E) > 0$ which means $F^\top E$ has rank k. Since columns of $F^\top E$ are in $\mathbb{R}^k$ this means $F^\top E$ has full column rank so we can find $y \in \text{rowspace}(V^\top)$ with $F^\top E y = z$.

Then since y is in the rowspace of $F^\top E$, a standard result (e.g., see Lemma A.4) is:

$$\|z\|_2 = \|F^\top E y\|_2 \geq \sigma_k(F^\top E)\|y\|_2 = c\|y\|_2 \tag{A.9}$$

But since F has orthonormal columns, $\|r\|_2 = \|Fz\|_2 = \|z\|_2$. This means $\|y\|_2 \leq \|r\|_2 / c \leq 1$.

Letting $u = Ey$, so far we've shown that $r = Fz = F(F^\top E y) = (FF^\top)u$, with $\|y\|_2 \leq 1$, where we note that $FF^\top = \Pi_R$ is the projection operator onto R. So $r = \Pi_R(u)$. But since the columns of E are orthonormal, we have $\|u\|_2 = \|Ey\|_2 = \|y\|_2 \leq 1$. $u = Ey$ is also in $S^\perp$, because the columns of E form a basis for $S^\perp$.

So to summarize, we found $u \in S^\perp$ such that $\Pi_R(u) = r$ and $\|u\| \leq 1$, as desired.

□

In the lemma above, we used a standard linear algebraic result that we include for completeness. This says that A cannot shrink vectors in its rowspace too much, where the shrinkage factor is given by the minimum singular value of A.

Lemma A.4. *Let $A \in \mathbb{R}^{m \times n}$. Let $r = \min(m, n)$. Then if $x \in$ rowspace(A), we have $\|Ax\|_2 \geq \sigma_r(A)\|x\|_2$.*

Proof. We bound the norm of x using the SVD. Consider the singular value decomposition (SVD) of A:

$$A = UDV^\top \tag{A.10}$$

Where $U \in \mathbb{R}^{m \times r}, D \in \mathbb{R}^{r \times r}, V^\top \in \mathbb{R}^{r \times n}$, where U and V have orthonormal columns, and $D = \text{diag}(\sigma_1, \ldots, \sigma_r)$ is a diagonal matrix with $\sigma_1 \geq \ldots \geq \sigma_r \geq 0$.

$$\begin{align}
\|Ax\|_2 &= \|UDV^\top x\|_2 && \text{[Definition of } r\text{]} \tag{A.11}\\
&= \|DV^\top x\|_2 && [U \in \mathbb{R}^{m \times r} \text{ has orthonormal columns}] \tag{A.12}\\
&\geq \sigma_r \|V^\top x\|_2 && [D \text{ is diagonal}] \tag{A.13}\\
&= \sigma_r \|x\|_2 && [\text{rows of } V^\top \text{ are orthonormal, } x \text{ is in rowspace}] \tag{A.14}\\
&= \sigma_r(A)\|x\|_2 && \tag{A.15}
\end{align}$$

Where for the fourth step, we used the fact that if $x \in \text{rowspace}(V^\top)$ and the rows of $V^\top$ are orthonormal, then $\|V^\top x\|_2 = \|x\|_2$. One way to see this is by writing $x = \sum_i \alpha_i v_i$, where v_i are rows of $V^\top$, and then noting that $V^\top x = (\alpha_1, \ldots, \alpha_r)$ and so x and $V^\top x$ have the same norm. □

We now prove Theorem 3.1, following the 3 steps outlined in the main text.

Proof of Theorem 3.1. Let $c = \cos\theta_k(R, S^\perp)$.

Because it makes the proof much easier, we will prove the contrapositive, and then convert back to the original theorem statement. We assume $\sqrt{L_{\text{ood}}(v_{ft}^t, B_{ft}^t)} \leq \Delta$, and will show that:

$$|(v_0^\top v_\star)^2 - (v_\star^\top v_\star)^2| \leq \frac{\Delta + \epsilon}{c} g_1(\|w\|_2)\sqrt{k} + \frac{(\Delta + \epsilon)^2}{c^2} g_2(\|w\|_2)k \tag{A.16}$$

Where g_1 and g_1 are non-negative polynomials we will bound in the proof.

We gave a basic outline of the proof in the main paper, and here we are just trying to be careful about capturing all the dependencies. We also give intuition for each step before diving into algebra (which we include for completeness).

Recall that in the overparameterized linear setting we assumed we have orthonormal B_0 with $\|B_0 - UB_\star\|_2 \leq \epsilon$ for some U. We note that the setup is rotationally symmetric so without loss of generality we can suppose $\|B_0 - UB_\star\|_2 \leq \epsilon$. This is because we can let $B'_\star = UB_\star$ and $v'_\star = Uv_\star$, and we have $w_\star = B_\star^\top v_\star = (UB_\star)^\top (Uv_\star)$, where $w_\star$ is the optimal classifier—so we can now write the entire proof in terms of $B'_\star$ and $v'_\star$.

Step 1: Show that $\|v_{ft}^t - v_\star\|_2 \leq \Delta/c$: We first give intuition and then dive into the math. The key insight is to use the fact that in 'many' directions B_{ft}^t and B_0 are the same (formally, for all $x \in S^\perp$, $B_{ft}^t x = B_0 x$). But B_0 and $B_\star$ are close by assumption, which means that B_{ft}^t and $B_\star$ are close in 'many' directions. Since we assumed in the contrapositive that ${v_{ft}^t}^\top B_{ft}^t$ and $v_\star^\top B_\star$ are close, we get that v_{ft}^t and $v_\star$ are close in 'many' directions. Because $S^\perp$ covers the rowspace of B_0, we get that 'many' is k, which is precisely the dimensionality of $v_\star$, so the two vectors v_{ft}^t and $v_\star$ must be close.

We now dive into the math. Since B_0 has orthogonal rows, B_0 has full column rank.

Let z be given by:

$$z = \frac{c}{\|v_{ft}^t - v_\star\|_2}(v_{ft}^t - v_\star) \tag{A.17}$$

We note that $\|z\|_2 = c$. Then, we can find $y \in R = \text{rowspace}(B_0)$ such that $B_0 y = z$ (since B_0 has full column-rank) and then $\|y\|_2 = \|z\|_2 = c$ (since B_0 has orthonormal rows).

Since $c = \cos\theta_k(R, S^\perp) > 0$, and $y \in R$ with $\|y\| = c$, from Lemma A.3 we can choose $x \in S^\perp$ with $\|x\|_2 \leq 1$ and $\Pi_R(x) = y$. Then, we have $B_0 x = z$.

From Proposition A.1, since $x \in S^\perp$, B_0 does not change in directions of x when fine-tuning so we have: $B_0 x = B_{ft}^t x$.

The claim now follows from simple algebraic manipulation, following the intuition we described. The algebra just captures what 'close' means and adds up the error terms.

$$\|v_{ft}^t - v_\star\|_2 = \frac{1}{c}(v_{ft}^t - v_\star)^\top\left(\frac{c(v_{ft}^t - v_\star)}{\|v_{ft}^t - v_\star\|_2}\right) \qquad \text{[Algebra]} \tag{A.18}$$

$$= \frac{1}{c}(v_{ft}^t - v_\star)^\top z \qquad \text{[Definition of } z\text{]} \tag{A.19}$$

$$= \frac{1}{c}(v_{ft}^t - v_\star)^\top B_0 x \qquad \text{[Since } B_0 x = z\text{]} \tag{A.20}$$

$$= \frac{1}{c}({v_{ft}^t}^\top B_0 x - v_\star^\top B_0 x) \qquad \text{[Algebra]} \tag{A.21}$$

$$= \frac{1}{c}({v_{ft}^t}^\top B_{ft}^t x - v_\star^\top B_0 x) \qquad [B_{ft}^t x = B_0 x \text{ since } x \in S^\perp] \tag{A.22}$$

$$= \frac{1}{c}({v_{ft}^t}^\top B_{ft}^t - v_\star^\top B_0) x \qquad \text{[Algebra]} \tag{A.23}$$

$$\leq \frac{1}{c}\|{v_{ft}^t}^\top B_{ft}^t - v_\star^\top B_0\|_2 \|x\|_2 \qquad \text{[Cauchy-Schwarz]} \tag{A.24}$$

$$\leq \frac{1}{c}\|{v_{ft}^t}^\top B_{ft}^t - v_\star^\top B_0\|_2 \qquad [\text{since } \|x\|_2 \leq 1] \tag{A.25}$$

$$\leq \frac{1}{c}\|{v_{ft}^t}^\top B_{ft}^t - v_\star^\top B_\star\|_2 + \frac{1}{c}\|v_\star^\top B_\star - v_\star^\top B_0\|_2 \qquad \text{[Triangle inequality]} \tag{A.26}$$

$$= \frac{1}{c}L_{\mathsf{ood}}(v_{ft}^t, B_{ft}^t) + \frac{1}{c}\|v_\star^\top B_\star - v_\star^\top B_0\|_2 \qquad [\text{definition of } L_{\mathsf{ood}}] \tag{A.27}$$

$$= \frac{1}{c}L_{\mathsf{ood}}(v_{ft}^t, B_{ft}^t) + \frac{1}{c}\sigma_{\max}(B_0 - B_\star)\|v_\star\|_2 \qquad \text{[definition of max singular value]} \tag{A.28}$$

$$= \frac{1}{c}L_{\mathsf{ood}}(v_{ft}^t, B_{ft}^t) + \frac{1}{c}\epsilon\|v_\star\|_2 \qquad [\text{since } \sigma_{\max}(B_0 - B_\star) \leq \epsilon] \tag{A.29}$$

$$\leq \frac{\Delta + \epsilon\|v_\star\|_2}{c} \qquad [\text{since } L_{\mathsf{ood}}(v_{ft}^t, B_{ft}^t) \leq \Delta] \tag{A.30}$$

$$\tag{A.31}$$

Which shows that $\|v_{ft}^t - v_\star\|_2 \leq (\Delta + \epsilon\|v_\star\|_2)/c$.

Step 2A: Show that $\|B_{ft}^t\|_F^2$ is small: The key insight is to take the trace on both sides of Proposition A.2, which bounds the Frobenius norm of B_{ft}^t and therefore the operator norm.

Rearranging Proposition A.2, we have:

$$B_{ft}^t {B_{ft}^t}^\top = B_0 B_0^\top + v_\star v_\star^\top - v_0 v_0^\top \tag{A.32}$$

Taking the trace everywhere, we get:

$$\mathrm{Tr}(B_{ft}^t {B_{ft}^t}^\top) = \mathrm{Tr}(B_0 B_0^\top) + \mathrm{Tr}(v_\star v_\star^\top) - \mathrm{Tr}(v_0 v_0^\top) \tag{A.33}$$

For any matrix A, $\mathrm{Tr}(AA^\top) = \|A\|_F^2$, and for a vector v the Frobenius norm is just the ℓ_2-norm, so $\mathrm{Tr}(vv^\top) = \|v\|_2^2$. So we have:

$$\|B_{ft}^t\|_F^2 = \|B_0\|_F^2 + \|v_\star\|_2^2 - \|v_0\|_2^2 \tag{A.34}$$

Squares are non-negative, so we get the inequality:

$$\|B_{ft}^t\|_F^2 \leq \|B_0\|_F^2 + \|v_\star\|_2^2 \tag{A.35}$$

Step 2B: Show that $\|B_0^\top v_\star\|_2^2 - \|{B_{ft}^t}^\top v_\star\|_2^2$ is small: This step doesn't involve much insight, and is standard peturbation analysis—we simply factor the difference of squares and bound each term.

First, we bound $\|{B^t_{ft}}^\top v^t_{ft} - {B^t_{ft}}^\top v_\star\|_2$:

$$\begin{align}
\|{B^t_{ft}}^\top v^t_{ft} - {B^t_{ft}}^\top v_\star\|_2 &\leq \sigma\text{max}(B^t_{ft})\|v^t_{ft} - v_\star\|_2 \tag{A.36}\\
&\leq \|B^t_{ft}\|_F\|v^t_{ft} - v_\star\|_2 \tag{A.37}\\
&\leq \sqrt{\|B_0\|_F^2 + \|v_\star\|_2^2}\|v^t_{ft} - v_\star\|_2 \tag{A.38}\\
&\leq \sqrt{\|B_0\|_F^2 + \|v_\star\|_2^2}\Big(\frac{\Delta + \epsilon\|v_\star\|_2}{c}\Big) \tag{A.39}
\end{align}$$

Next, we bound $\|B_0^\top v_\star - {B^t_{ft}}^\top v_\star\|_2$:

$$\begin{align}
\|B_0^\top v_\star - {B^t_{ft}}^\top v_\star\|_2 &\leq \|B_0^\top v_\star - B_\star^\top v_\star\|_2 + \|B_\star^\top v_\star - {B^t_{ft}}^\top v_\star\|_2 \tag{A.40}\\
&\leq \sigma_{\max}(B_0 - B_\star)\|v_\star\|_2 + \|B_\star^\top v_\star - {B^t_{ft}}^\top v_\star\|_2 \tag{A.41}\\
&\leq \epsilon\|v_\star\|_2 + \|B_\star^\top v_\star - {B^t_{ft}}^\top v_\star\|_2 \tag{A.42}\\
&\leq \epsilon\|v_\star\|_2 + \|B_\star^\top v_\star - {B^t_{ft}}^\top v^t_{ft}\|_2 + \|{B^t_{ft}}^\top v^t_{ft} - {B^t_{ft}}^\top v_\star\|_2 \tag{A.43}\\
&\leq \epsilon\|v_\star\|_2 + \Delta + \sqrt{\|B_0\|_F^2 + \|v_\star\|_2^2}\Big(\frac{\Delta + \epsilon\|v_\star\|_2}{c}\Big) \tag{A.44}\\
&=: \Delta_2 \tag{A.45}
\end{align}$$

Finally, we bound $|\|B_0^\top v_\star\|_2^2 - \|{B^t_{ft}}^\top v_\star\|_2^2|$, using the identity:

$$\begin{align}
|\|u\|_2^2 - \|v\|_2^2| &= |(u-v)^\top(u+v)| \tag{A.46}\\
&\leq \|u-v\|_2\|u+v\|_2 \tag{A.47}\\
&\leq \|u-v\|_2(2\|u\|_2 + \|u-v\|_2) \tag{A.48}
\end{align}$$

Applying this:

$$\begin{align}
|\|B_0^\top v_\star\|_2^2 - \|{B^t_{ft}}^\top v_\star\|_2^2| &\leq \|B_0^\top v_\star - {B^t_{ft}}^\top v_\star\|_2(2\|B_0^\top v_\star\|_2 + \|B_0^\top v_\star - {B^t_{ft}}^\top v_\star\|_2) \tag{A.49}\\
&\leq \Delta_2(2\|B_0^\top v_\star\|_2 + \Delta_2) \tag{A.50}\\
&\leq \Delta_2(2\|B_\star^\top v_\star\|_2 + 2\|B_0^\top v_\star - B_\star^\top v_\star\|_2 + \Delta_2) \tag{A.51}\\
&\leq \Delta_2(2\|w_\star\|_2 + 2\epsilon\|v_\star\|_2 + \Delta_2) \tag{A.52}\\
&=: \Delta_3 \tag{A.53}
\end{align}$$

Step 3: Use Proposition A.2 to show v_0 and $v_\star$ must be close: The key insight is that we start from Proposition A.2, and left and right multiply by $v_\star$, after that we use the previous steps and do some some standard perturbation analysis.

We start from Proposition A.2:

$$v_0 v_0^\top - B_0 B_0^\top = v^t_{ft}{v^t_{ft}}^\top - B^t_{ft}{B^t_{ft}}^\top \tag{A.54}$$

The key step is to left multiply both sides by $v_\star^\top$ and right multiply both sides by $v_\star$ to get:

$$(v_0^\top v_\star)^2 - \|B_0^\top v_\star\|_2^2 = ({v^t_{ft}}^\top v_\star)^2 - \|{B^t_{ft}}^\top v_\star\|_2^2 \tag{A.55}$$

Rearranging, and then using Equation A.49, we get:

$$|({v^t_{ft}}^\top v_\star)^2 - (v_0^\top v_\star)^2| = |\|{B^t_{ft}}^\top v_\star\|_2^2 - \|B_0^\top v_\star\|_2^2| \leq \Delta_3 \tag{A.56}$$

This is close to what we want, except we have $({v^t_{ft}}^\top v_\star)^2$ on the LHS instead of $(v_\star^\top v_\star)^2$. We previously showed that v^t_{ft} and $v_\star$ are close, in Step 1, so with some algebra we can bound the difference between $({v^t_{ft}}^\top v_\star)^2$ and $(v_\star^\top v_\star)^2$:

$$\begin{align}
|({v^t_{ft}}^\top v_\star)^2 - (v_\star^\top v_\star)^2| &= |({v^t_{ft}}^\top v_\star - v_\star^\top v_\star)^\top({v^t_{ft}}^\top v_\star + v_\star^\top v_\star)| \tag{A.57}\\
&= |({v^t_{ft}}^\top v_\star - v_\star^\top v_\star)^\top[2v_\star^\top v_\star + ({v^t_{ft}}^\top v_\star - v_\star^\top v_\star)]| \tag{A.58}\\
&= |(v_\star^\top(v^t_{ft} - v_\star))^\top[2v_\star^\top v_\star + (v_\star^\top(v^t_{ft} - v_\star))]| \tag{A.59}\\
&\leq \|v^t_{ft} - v_\star\|_2\|v_\star\|_2^2[2\|v_\star\|_2 + \|v^t_{ft} - v_\star\|_2] \tag{A.60}\\
&= (\Delta/c)\|v_\star\|_2^2(2\|v_\star\|_2 + (\Delta/c)) := \Delta_4 \tag{A.61}
\end{align}$$

Above, from the third line to the fourth line, we used triangle inequality and Cauchy-Schwarz.

So finally, by triangle-inequality we can now bound $|(v_\star^\top v_\star)^2 - (v_0^\top v_\star)^2|$:

$$|(v_\star^\top v_\star)^2 - (v_0^\top v_\star)^2| \leq |(v_\star^\top v_\star)^2 - ({v_{ft}^t}^\top v_\star)^2| + |({v_{ft}^t}^\top v_\star)^2 - (v_0^\top v_\star)^2| \tag{A.62}$$

$$\leq \Delta_4 + \Delta_3 \tag{A.63}$$

Wrap up i.e., writing out $\Delta_4 + \Delta_3$ explicitly: This is basically the bound we want, but we would like to express Δ_3, Δ_4 in terms of Δ and ϵ. Note that this step has no insight, and is just algebra—we include the details for reference and verifiability. We recall:

$$\Delta_4 = (\Delta/c)\|v_\star\|_2^2(2\|v_\star\|_2 + (\Delta/c)) \tag{A.64}$$

$$\Delta_3 = \Delta_2(2\|w_\star\|_2 + 2\epsilon\|v_\star\|_2 + \Delta_2) \tag{A.65}$$

$$\Delta_2 = \epsilon\|v_\star\|_2 + \Delta + \sqrt{\|B_0\|_F^2 + \|v_\star\|_2^2}\Big(\frac{\Delta + \epsilon\|v_\star\|_2}{c}\Big) \tag{A.66}$$

Since B_0 has orthogonal rows (by assumption), $B_0^\top$ has orthogonal columns, so $\|w_\star\|_2 = \|B_0^\top v_\star\|_2 = \|v_\star\|_2$. In addition, since B_0 has k orthogonal rows, $\|B_0\|_F = \sqrt{k}$. We also note that $\sqrt{\|B_0\|_F^2 + \|v_\star\|_2^2} \leq \|B_0\|_F + \|v_\star\|_2 = \sqrt{k} + \|w_\star\|_2$. Since $c \leq 1$, we have:

$$\epsilon\|v_\star\|_2 + \Delta \leq \Big(\frac{\Delta + \epsilon\|v_\star\|_2}{c}\Big) \tag{A.67}$$

So for Δ_2, up to constant factors we can ignore the $\epsilon\|v_\star\|_2 + \Delta$ term—this means we get:

$$\Delta_2 \leq O\Big((\sqrt{k} + \|w_\star\|_2)\Big(\frac{\Delta + \epsilon\|w_\star\|_2}{c}\Big)\Big) \tag{A.68}$$

Using the fact that $\sqrt{k} + \|w_\star\|_2 \leq \sqrt{k}(1 + \|w_\star\|)$ we get:

$$\Delta_2 \leq O\Big(\sqrt{k}(1 + \|w_\star\|)\Big(\frac{\Delta + \epsilon\|w_\star\|_2}{c}\Big)\Big) \tag{A.69}$$

Then since $\Delta + \epsilon\|w_\star\|_2 \leq (1 + \|w_\star\|_2)(\Delta + \epsilon)$, we get:

$$\Delta_2 \leq O\Big(\sqrt{k}(1 + \|w_\star\|)^2\Big(\frac{\Delta + \epsilon}{c}\Big)\Big) \tag{A.70}$$

Now for Δ_3, first note that $\epsilon \leq 2$, since $B_\star$ and B_0 have orthogonormal rows so $\|B_\star - B_0\|_2 \leq 2$. This means that $\epsilon\|w_\star\|_2 \leq \|w_\star\|_2$, so Δ_3 simplifies to:

$$\Delta_3 \leq O(\Delta_2(\|w_\star\|_2 + \Delta_2)) = O(\Delta_2\|w_\star\|_2 + \Delta_2) \tag{A.71}$$

Substituting the bound for Δ_2 into Δ_3, we get:

$$\Delta_3 \leq O\Big(\sqrt{k}\|w_\star\|_2(1 + \|w_\star\|)^2\Big(\frac{\Delta + \epsilon\|w_\star\|_2}{c}\Big) + k(1 + \|w_\star\|)^4\Big(\frac{\Delta + \epsilon\|w_\star\|_2}{c}\Big)^2\Big) \tag{A.72}$$

For Δ_4, we get:

$$\Delta_4 \leq O\Big(\|w_\star\|_2^3\frac{\Delta}{c} + \|w_\star\|_2(\frac{\Delta}{c})\Big) \tag{A.73}$$

Since $\Delta/c \leq (\Delta + \epsilon)/c$ and $\|w_\star\|_2^2 \leq (1 + \|w_\star\|_2)^2$ we have for the final error $\Delta_3 + \Delta_4$:

$$\Delta_3 + \Delta_4 \leq \sqrt{k}w(1 + \|w_\star\|_2^2)^2\Big(\frac{\Delta + \epsilon}{c}\Big) + k(1 + \|w_\star\|_2^2)^4\Big(\frac{\Delta + \epsilon}{c}\Big)^2 \tag{A.74}$$

Wrap up i.e., taking the contrapositive: So we've shown that if $\sqrt{L_{\text{ood}}(v_{ft}^t, B_{ft}^t)} \leq \Delta$, then:

$$|(v_\star^\top v_\star)^2 - (v_0^\top v_\star)^2| \leq \frac{\Delta + \epsilon}{c}w(1 + \|w_\star\|_2^2)^2\sqrt{k} + \frac{(\Delta + \epsilon)^2}{c^2}(1 + \|w_\star\|_2^2)^4 k \tag{A.75}$$

We'd like to flip this around: suppose $|(v_\star^\top v_\star)^2 - (v_0^\top v_\star)^2| \geq \varphi^2$ for some φ. To lower bound $L_{\text{ood}}(v_{ft}^t, B_{ft}^t)$, we simply take the contrapositive of what we have proved. Let Δ be given by:

$$\Delta = \min\Big(\frac{c}{w(1 + \|w_\star\|_2^2)^2\sqrt{k}}\varphi^2, \frac{c}{\sqrt{(1 + \|w_\star\|_2^2)^4 k}}\varphi\Big) - \epsilon \tag{A.76}$$

In this case with some algebra, we can show that:

$$|(v_\star^\top v_\star)^2 - (v_0^\top v_\star)^2| \geq \varphi^2 \geq \frac{\Delta + \epsilon}{c} w(1 + \|w_\star\|_2^2)^2 \sqrt{k} + \frac{(\Delta + \epsilon)^2}{c^2}(1 + \|w_\star\|_2^2)^4 k \tag{A.77}$$

To see this, we bound each of the terms in the RHS separately using our definition of Δ. Then, from the contrapositive of what we proved (compare with Equation A.75, we get:

$$\sqrt{L_{\mathsf{ood}}(v_{ft}^t, B_{ft}^t)} \geq \Delta \tag{A.78}$$

Finally, we can massage Δ to combine terms and make it look slightly nicer:

$$\Delta \geq \frac{c}{\sqrt{k}} \frac{\min(\varphi, \varphi^2 / \|w_\star\|_2)}{(1 + \|w_\star\|_2)^2} - \epsilon \tag{A.79}$$

For even more interpretability, if $\|w\|_2 = 1$ and φ is bounded above by some constant, then you can think of Δ as approximately $\frac{c}{\sqrt{k}}\varphi^2 - \epsilon$. This completes the proof. □

A.3 LP VS FT IN THE GAUSSIAN SETTING (OOD)

We now prove Theorem 3.2, which compared linear probing and fine-tuning in the linear overparameterized setting, when the covariates were Gaussian.

Restatement of Theorem 3.2. *In the linear overparameterized setting, suppose the training data is Gaussian. and recall that ϵ is the error in the pretrained feature extractor. Then as the feature extractor error ϵ goes to 0, linear probing does much better than fine-tuning OOD:*

$$\frac{L_{\mathsf{ood}}(v_{\mathsf{lp}}^\infty, B_0)}{L_{\mathsf{ood}}(v_{\mathsf{ft}}(t), B_{\mathsf{ft}}(t))} \xrightarrow{p} 0, as \;\; \epsilon \to 0 \tag{A.80}$$

This holds for all times t for FT (and therefore also for the limit $v_{\mathsf{ft}}^\infty, B_{\mathsf{ft}}^\infty$) and the LP iterates converge to $v_{\mathsf{lp}}^\infty, B_0$ as a result of the gradient flow on a convex problem.

At a high level, we use Theorem 3.1 to lower bound the OOD error of fine-tuning, and then upper bound the OOD error of linear probing.

To lower bound the OOD error of fine-tuningwe need to lower bound $\cos\theta_k(R, S^\perp)$ and Head-Error$(v_0, v_\star)$. We reduce lower bounding $\cos\theta_k(R, S^\perp)$ to a Random Matrix Theory of bounding the min and max singular values of matrices involving B_0 and $S^\perp$. Lower bounding Head-Error$(v_0, v_\star)$ is a straightforward anti-concentration result (showing that a sample from a Gaussian is not too close to 0).

We first bound $\cos\theta_k(R, S)$ and then the bound for $\cos\theta_k(R, S^\perp)$ follows. We show that $\cos\theta_k(R, S) > 0$ almost surely (with probability 1), and moreover give a quantitative lower bound for $\cos\theta_k(R, S)$.

Lemma A.5. *Fix a k dimensional subspace R. Let S be a subspace spanned by $x_1, \ldots, x_n \in \mathbb{R}^d$ where x_i are sampled independently from $N(0, \sigma^2 I)$, with $k < n < d$. Then with probability at least $1 - \delta$,*

$$\cos\theta_k(R, S) \geq \frac{\sqrt{n} - \sqrt{k} - \sqrt{2\log\frac{1}{\delta}}}{\sqrt{d \log\frac{2d}{\delta}}} \tag{A.81}$$

In addition, we get that $\cos\theta_k(R, S) > 0$ almost surely (with probability 1).

If $n \geq 5k$ and $n \geq 10\log\frac{1}{\delta}$, then we get:

$$\cos\theta_k(R, S) \geq O\Big(\sqrt{\frac{n}{d\log\frac{2d}{\delta}}}\Big) \tag{A.82}$$

Recall that big-oh notation here means that the RHS is true for some universal constant (independent of any other problem parameters).

Proof. We note that since $x_1, \ldots, x_n$ are sampled from $N(0, \sigma^2 I_d)$, which is rotationally invariant, every rotation of the subspace spanned by the n points has equal probability. In other words, we are choosing a uniformly random n dimensional subspace (formally: this is a uniform random measure over the Grassmannian manifold).

Note that principal angles are invariant if we rotate R and S by the same rotation matrix U. This symmetry means that we can fix S and instead consider R to be a uniform random k dimensional subspace on the Grassmannian manifold. Without loss of generality, we can also fix S to be the first k standard basis vectors $((1, 0, 0, \ldots, 0), (0, 1, 0, \ldots, 0), \ldots, (0, 0, 0, \ldots, 1))$.

Equivalently, let M_R be a d-by-k matrix, where each column is sampled independently from $N(0, I_d)$ (the σ^2 makes no difference to the subspace)—since the columns of M_R span a uniformly random k-dimensional subspace, so we can let R be range of M_R. This is equivalent to sampling each entry of M_R from $N(0, 1)$.

Let $c = \cos\theta_k(R, S)$. We note that c can be written as (this is a simple transformation of Definition 3.1):

$$c = \min_{r \in R, \|r\|_2 \leq 1} \|\Pi_S(r)\|_2 \tag{A.83}$$

Now, any $r \in R$ can be written as $r = M_R\lambda$ for some $\lambda \in \mathbb{R}^k$. We first show that $\|\lambda\|_2$ cannot be much smaller than $\|r\|_2$. This is because:

$$\|r\|_2 = \|M_R\lambda\|_2 \leq \sigma_{\max}(M_R)\|\lambda\|_2 \tag{A.84}$$

So this gives us:

$$\|\lambda\|_2 \geq \frac{\|r\|_2}{\sigma_{\max}(M_R)} \tag{A.85}$$

So every $r \in R$ can be written as $M_R\lambda$ where $\|\lambda\|_2$ is lower bounded as above.

We now simplify the definition of c, starting from Equation A.83.

$$\begin{aligned}
c &= \min_{r \in R, \|r\|_2 \leq 1} \|\Pi_S(r)\|_2 && \text{(A.86)}\\
&\geq \min_{\|\lambda\| \geq 1/\sigma_{\max}(M_R)} \|\Pi_S M_R \lambda\|_2 && \text{(A.87)}\\
&\geq \min_{\|\lambda\| \geq 1/\sigma_{\max}(M_R)} \sigma_{\min}(\Pi_S M_R)\|\lambda\|_2 && \text{(A.88)}\\
&= \frac{\sigma_{\min}(\Pi_S M_R)}{\sigma_{\max}(M_R)} && \text{(A.89)}
\end{aligned}$$

So now we want to lower bound the ratio of two random matrices. We note that $\Pi_S M_R$ is a matrix of size (n, k) with each entry sampled independently from $N(0, 1)$ (this is because Π_S simple selects the first n rows of M_R). M_R is a matrix of size (d, k) with each entry sampled independently from $N(0, 1)$. So standard matrix concentration results (Tropp, 2015; Absil et al., 2006) bound the singular values, and we get that with probability at least $1 - \delta$:

$$\sigma_{\min}(\Pi_S M_R) \geq \sqrt{n} - \sqrt{k} - \sqrt{2 \log \frac{1}{\delta}} \tag{A.90}$$

$$\sigma_{\max}(M_R) \leq \sqrt{d \log \frac{2d}{\delta}} \tag{A.91}$$

Note that we can use alternate bounds for $\sigma_{\min}$ that are sometimes tighter. So for the ratio of the two, we get that with probability at least $1 - \delta$, we have:

$$c \geq \frac{\sigma_{\min}(\Pi_S M_R)}{\sigma_{\max}(M_R)} \geq \frac{\sqrt{n} - \sqrt{k} - \sqrt{2 \log \frac{2}{\delta}}}{\sqrt{d \log \frac{2d}{\delta}}} \tag{A.92}$$

For interpretability, ignoring log factors this is approximately:

$$c \gtrapprox \frac{\sqrt{n} - \sqrt{k}}{\sqrt{d}} \tag{A.93}$$

The result when $n \geq 5k$ and $n \geq 10 \log \frac{2}{\delta}$ follows with simple algebra.

For the result where we show $\cos \theta_k(R, S) > 0$ almost surely, we recall that $\Pi_S M_R$ is a matrix of size (n, k) with each entry sampled independently from $N(0, 1)$. Then applying Lemma 3 in Xie et al. (2021a), we get that $\sigma_{\min}(\Pi_S M_R) > 0$ almost surely. Since $\sigma_{\max}(M_R)$ is finite, this gives us $\cos \theta_k(R, S) > 0$ almost surely. □

If the training data is Gaussian then the span of the training data S is a random n-dimensional subspace with $n < d$. So the orthogonal complement $S^\perp$ is a random $d - n$ dimensional subspace and the same argument as Lemma A.5 applies.

Corollary A.1. *In the linear overaparameterized setting, under the Gaussian data assumption, let $R = range(B_0)$. Then with probability at least $1 - \delta$, where the probability is over the fine-tuning examples:*

$$\cos \theta_k(R, S^\perp) \geq \frac{\sqrt{d-n} - \sqrt{k} - \sqrt{2 \log \frac{1}{\delta}}}{\sqrt{d \log \frac{2d}{\delta}}} \tag{A.94}$$

In addition, we get that $\cos \theta_k(R, S^\perp) > 0$ almost surely (with probability 1).

If $d - n \geq 5k$ and $d - n \geq 10 \log \frac{1}{\delta}$, then we have:

$$\cos \theta_k(R, S^\perp) \geq \sqrt{\frac{d-n}{d \log \frac{2d}{\delta}}} \tag{A.95}$$

We now bound Head-Error$(v_0, v_\star)$. Note that if the head is initialized as $v_0 = 0$, then Head-Error$(v_0, v_\star) = \|v_\star\|_2^2 = \|w_\star\|_2^2$. In practice, the head is usually initialized randomly, for example normally distributed. Intuitively, the head error is still high because we do not know which direction the head is pointing in, so most of the time the initial (randomly sampled) head will be pointing in the wrong direction. If $v_0 \sim N(0, \sigma^2 I)$ can show that for any σ^2, the head error will still typically be at least $\Omega(\|v_\star\|_2)$ This is an illustrative result, one can show similar results for other random initializations as well.

We first prove an anti-concentration lemma, which says that if u is univariate Gaussian, then it cannot be too close to any particular constant a, no matter how the variance of the Gaussian is chosen.

Lemma A.6. *For some universal constant c, given $a > 0$, for all ν^2 if $u \sim N(0, \nu^2)$ then for all $0 \leq \delta \leq 1$:*

$$P(|u - a| \leq c\delta a) \leq \delta \tag{A.96}$$

Proof. Consider δ such that $\delta \leq 1/10$. Then for all u with $|u - a| \leq \delta a$, we have $u \geq 9a/10$. For all $u \geq 9a/10$, the density $f(u)$ is upper bounded (from the formula for the density of a Gaussian random variable) by:

$$f(u) \leq O(\frac{1}{v} \exp \frac{-9^2 a^2}{2 \cdot 10^2 v^2}) \leq \frac{1}{v}(1 - \frac{9^2 a^2}{2 \cdot 10^2 v^2}) \tag{A.97}$$

We can maximize this explicitly (e.g., use Mathematica) and we get for some universal constant $c' \geq 10$ (it is OK to choose a larger universal constant than needed):

$$f(u) \leq \frac{c'}{a} \tag{A.98}$$

Since the density is less than $1/a$ and if $|u - a| \leq \delta a$ the size of the interval is $2\delta a$, we get for all $\delta \leq 1/10$:

$$P(|u - a| \leq \delta a) \leq \frac{2c'\delta a}{a} = 2c'\delta \tag{A.99}$$

Now, we substitute $\delta' = 2c'\delta$. We get for all $\delta' \leq 2c'/10$:

$$P(|u - a| \leq \frac{1}{2c'}\delta' a) \leq \delta' \tag{A.100}$$

Since $c' \geq 10$, $2c'/10 \geq 1$, so the statement is true for all $0 \leq \delta' \leq 1$. □

We now bound the error in the head if the initialization is Gaussian. This bound holds for all initialization variances σ^2. Similar bounds can be shown for other (non-Gaussian) head initializations using similar anti-concentration arguments.

Lemma A.7. *For all σ, if $v_0 \sim N(0, \sigma^2 I_k)$, for some universal constant c, we have with probability at least $1 - \delta$:*

$$(\textit{Head-Error}(v_0, v_\star))^2 = |(v_0^\top v_\star)^2 - (v_\star^\top v_\star)^2| > c\delta(v_\star^\top v_\star)^2 \tag{A.101}$$

Proof. First note that Head-Error$(v_0, v_\star) =$ Head-Error$(-v_0, v_\star)$ and v_0 is symmetric around 0 (v_0 and $-v_0$ have the same probability). So without loss of generality, we can suppose that $v_0^\top v_\star \geq 0$. **Suffices to bound** $|v_0^\top v_\star - v_\star^\top v_\star|$: We decompose the error:

$$|(v_0^\top v_\star)^2 - (v_\star^\top v_\star)^2| = |v_0^\top v_\star - v_\star^\top v_\star|(|v_0^\top v_\star + v_\star^\top v_\star|) \tag{A.102}$$

$$\geq |v_0^\top v_\star - v_\star^\top v_\star|(v_\star^\top v_\star)| \tag{A.103}$$

So we bound $|v_0^\top v_\star - v_\star^\top v_\star|$.

$v_0^\top v_\star$ **is normally distributed**: We note that $v_0^\top v_\star$ is distributed as:

$$v_0^\top v_\star \sim N(0, \sigma^2 v_\star^\top v_\star) \tag{A.104}$$

In other words, a normal with mean 0, and *variance* $\sigma_1^2 = \sigma^2 v_\star^\top v_\star$, and therefore standard deviation $\sigma_1 = \sigma\sqrt{v_\star^\top v_\star}$.

Apply Gaussian anti-concentration lemma: Then, from Lemma A.6, we have for some universal constant c that with probability at least $1 - \delta$:

$$|v_0^\top v_\star - v_\star^\top v_\star| \geq c\delta v_\star^\top v_\star \tag{A.105}$$

So substituting this back into Equation A.102, we get the desired result::

$$|(v_0^\top v_\star)^2 - (v_\star^\top v_\star)^2| > c\delta(v_\star^\top v_\star)^2 \tag{A.106}$$

□

Lemma A.8. *In the overparameterized linear setting, if $d \geq n \geq 5k$ and $n \geq 10 \log \frac{1}{\delta}$ then with probability at least $1 - \delta$, the linear probing OOD error is upper bounded by:*

$$\sqrt{L_{\mathsf{ood}}(v_{\mathsf{lp}}^\infty, B_0)} \leq O\Big(\frac{d}{n} \log{(\frac{d}{\delta})} \epsilon \|w_\star\|_2\Big) \tag{A.107}$$

Without these conditions on d, n, k, we have for every $\delta > 0$, there exists some $c_\delta > 0$ such that with probability $1 - \delta$:

$$\sqrt{L_{\mathsf{ood}}(v_{\mathsf{lp}}^\infty, B_0)} \leq c_\delta \epsilon \|w_\star\|_2 \tag{A.108}$$

Proof. Suppose $n > k$. Let X be a matrix of shape (n, d), where the i-th row contains the i-th data point (which we assumed was sampled from $N(0, I_d)$.

Before proceeding with the proof, we make a quick note that $BX^\top XB^\top$ is invertible almost surely. Reason: X is a Gaussian random matrix with each entry sampled independently from $N(0, \sigma^2)$, with shape (n, d). B is a matrix of shape (k, d) with orthogonal rows, so $BX^\top$ is a matrix of shape (k, n) with each entry sampled independently from $N(0, \sigma^2)$. Since $n > k$ this has rank k almost surely, and $BX^\top XB^\top$ has the same rank (k) and is therefore full rank, which means it is invertible.

In other words, there is a unique global minimum (minimizing over v) to the loss optimized by linear-probing:

$$\arg\min_v \|XB_0^\top v - XB_\star^\top v_\star\|_2^2 = (B_0 X^\top X B_0^\top)^{-1} B_0 X^\top X B_\star^\top v_\star \tag{A.109}$$

We can see this by noting that the loss function on the LHS is strongly convex in v since $BX^\top XB^\top$ is invertible. Then, gradient flow converges to the unique minimizer on the RHS, so:

$$v_{\mathsf{lp}}^\infty = (B_0 X^\top X B_0^\top)^{-1} B X^\top X B_\star^\top v_\star \tag{A.110}$$

We now bound the square-root OOD error (taking the square root makes it easier to apply triangle inequalities), starting with the definition:

$$\begin{aligned}
\sqrt{L_{\mathsf{ood}}(v_{\mathsf{lp}}^\infty, B_0)} &= \|B_\star^\top v_\star - B_0^\top v_{\mathsf{lp}}^\infty\|_2 && \text{(A.111)}\\
&\le \|(B_\star^\top v_\star - B_0^\top v_\star) + (B_0^\top v_\star - B_0 v_{\mathsf{lp}}^\infty)\|_2 && \text{(A.112)}\\
&\le \underbrace{\|B_\star^\top v_\star - B_0^\top v_\star\|_2}_{(1)} + \underbrace{\|B_0^\top v_\star - B_0 v_{\mathsf{lp}}^\infty)\|_2}_{(2)} && \text{(A.113)}
\end{aligned}$$

We bound each term on the RHS of the last line. For term (1):

$$\begin{aligned}
\|B_\star^\top v_\star - B_0^\top v_\star\|_2 &\le \sigma_{\max}(B_\star - B_0)\|v_\star\|_2 && \text{(A.114)}\\
&\le \epsilon\|w_\star\|_2 && \text{(A.115)}
\end{aligned}$$

Let $\Sigma = X^\top X$. For term (2), we first subtitute v_{lp}^∞ and do some algebra to get:

$$\begin{aligned}
\|B_0^\top v_\star - B_0 v_{\mathsf{lp}}^\infty)\|_2 &= \|B_0^\top (B_0 \Sigma B_0^\top)^{-1} B_0 \Sigma (B_0 - B_\star) v_\star\|_2 && \text{(A.116)}\\
&\le \sigma_{\max}(B_0^\top (B_0 \Sigma B_0^\top)^{-1} B_0 \Sigma) \sigma_{\max}(B_0 - B_\star)\|w_\star\|_2 && \text{(A.117)}\\
&\le \sigma_{\max}(B_0^\top (B_0 \Sigma B_0^\top)^{-1} B_0 \Sigma)\epsilon\|w_\star\|_2 && \text{(A.118)}\\
&\le \sigma_{\max}(B_0)^2 \sigma_{\max}(\Sigma) \frac{1}{\sigma_{\min}(B_0 \Sigma B_0^\top)} \epsilon\|w_\star\|_2 && \text{(A.119)}\\
&\le \frac{\sigma_{\max}(B_0)^2 \sigma_{\max}(X)^2}{\sigma_{\min}(XB^\top)^2} \epsilon\|w_\star\|_2 && \text{(A.120)}\\
& && \text{(A.121)}
\end{aligned}$$

So it suffices to bound the max singular value in the RHS. Since B_0 has orthonormal rows, $\sigma_{\max}(B_0) = 1$. For the other singular values, we do some standard random matrix theory (Tropp, 2015; Absil et al., 2006). We get with probability at least $1 - \delta$:

$$\sigma_{\max}(X)^2 \le O(d \log \frac{d}{\delta}) \tag{A.122}$$

$$\sigma_{\min}(XB^\top) \ge \sqrt{n} - \sqrt{k} - \sqrt{2 \log \frac{1}{\delta}} \tag{A.123}$$

We assumed $n \ge 5k$ and $n \ge 10 \log \frac{1}{\delta}$, then we get:

$$\sigma_{\min}(XB^\top) \ge O(\sqrt{n}) \tag{A.124}$$

Substituting back into Equation A.116 we get:

$$\|B_0^\top v_\star - B_0 v_{\mathsf{lp}}^\infty)\|_2 \le O\Big(\frac{d}{n} \log{(\frac{d}{\delta})} \epsilon\|w_\star\|_2\Big) \tag{A.125}$$

Substituting into equation A.111 and noting that $n \le d$, we have:

$$\sqrt{L_{\mathsf{ood}}(v_{\mathsf{lp}}^\infty, B_0)} \le O\Big(\frac{d}{n} \log{(\frac{d}{\delta})} \epsilon\|w_\star\|_2\Big) \tag{A.126}$$

Which completes the proof of the first part.

For the second part, where we want to show $\sqrt{L_{\mathsf{ood}}(v_{\mathsf{lp}}^\infty, B_0)} \le c_\delta \epsilon \|w_\star\|_2$ without the extra conditions on n, k, we use the fact that $\sigma_{\max}(X)$ is upper bounded almost surely, and $\sigma_{\min}(XB^\top) > 0$ almost surely (from Lemma 3 in Xie et al. (2021a), which only requires absolute continuity). This implies that for some $c_\delta > 0$ that can depend on δ, with probability at least $1 - \delta$:

$$\frac{\sigma_{\max}(B_0)^2 \sigma_{\max}(X)^2}{\sigma_{\min}(XB^\top)^2} \le c_\delta \tag{A.127}$$

which means that $\sqrt{L_{\mathsf{ood}}(v_{\mathsf{lp}}^\infty, B_0)} \le c_\delta \epsilon \|w_\star\|_2$, as desired. □

We now prove Theorem 3.2. This shows that the ratio of the OOD errors of linear probing over fine-tuning goes to 0 as the representation error ϵ goes to 0. In the next Proposition, we also give a stronger non-asymptotic result, where we give a particular threshold T (in terms of n, k, d), such that if $\epsilon < T$ then linear probing does better than fine-tuning.

Proof of Theorem 3.2. **Combining the fine-tuning bounds**: From Theorem 3.1, we have:

$$\sqrt{L_{\mathsf{ood}}(v_{ft}^t(t), B_{\mathsf{ft}}(t))} \geq \frac{\cos\theta_k(R, S^{\perp})}{\sqrt{k}} \frac{\min(\varphi, \varphi^2/\|w_\star\|_2)}{(1+\|w_\star\|_2)^2} - \epsilon \tag{A.128}$$

Here φ^2 is the head-error $|(v_\star^\top v_\star)^2 - (v_0^\top v_\star)^2|$. From Corollary A.1 we have $\cos\theta_k(R, S^{\perp}) > 0$ almost surely. From Lemma A.7 we have $\varphi^2 > 0$ almost surely. So for any $\delta > 0$, there exists some $c'_\delta > 0$ such that with probability at least $1 - \delta$:

$$\sqrt{L_{\mathsf{ood}}(v_{ft}^t(t), B_{\mathsf{ft}}(t))} \geq c'_\delta - \epsilon \tag{A.129}$$

Linear-probing bound: From Lemma A.8, for every $\delta > 0$, there exists some $c_\delta > 0$ such that with probability $1 - \delta$:

$$\sqrt{L_{\mathsf{ood}}(v_{\mathsf{lp}}^\infty, B_0)} \leq c_\delta \epsilon \|w_\star\|_2 \tag{A.130}$$

Ratio converges to 0 in probability: We simply take the ratio of the OOD errors of linear-probing and fine-tuning that we wrote above. Notice that we have an *upper bound* for linear-probing and a *lower bound* for fine-tuning. We get that with probability at least $1 - 2\delta$:

$$\frac{\sqrt{L_{\mathsf{ood}}(v_{\mathsf{lp}}^\infty, B_0)}}{\sqrt{L_{\mathsf{ood}}(v_{ft}^t, B_{\mathsf{ft}})}} \leq O\Big(\frac{c_\delta \epsilon \|w_\star\|_2}{c'_\delta - \epsilon}\Big) \tag{A.131}$$

Notice that for every failure probability δ, as $\epsilon \to 0$, the RHS goes to 0. Since the LHS is always non-negative, this implies convergence in probability to 0. This implies convergence in probability to 0 for the squared of the LHS as well, which is what we wanted to show. □

We can actually show a more precise, quantitative version of Theorem 3.2, which we do below.

Proposition A.1. *In the linear overparameterized setting, assuming Gaussian covariates, let* $\|w_\star\|_2$ *be a fixed constant. Then for all* $w_\star, B_0, n, d, k, \epsilon$*, if* $d - n, n \geq 5k$*,* $d - n, n \geq 10 \log \frac{1}{\delta}$*, and if* $\epsilon = \|B_0 - B_\star\|_2$ *is small so that:*

$$\epsilon \leq O\Big(\delta \sqrt{\frac{n^2(d-n)}{kd^3} \frac{1}{(\log \frac{d}{\delta})^3}}\Big) \tag{A.132}$$

Then we have for all times t*:*

$$L_{\mathsf{ood}}(v_{\mathsf{lp}}^\infty, B_0) < L_{\mathsf{ood}}(v_{ft}^t, B_{ft}^t) \tag{A.133}$$

Proof. **Combining the fine-tuning bounds**: From Theorem 3.1, we have:

$$\sqrt{L_{\mathsf{ood}}(v_{ft}^t(t), B_{ft}^t(t))} \geq \frac{\cos\theta_k(R, S^{\perp})}{\sqrt{k}} \frac{\min(\varphi, \varphi^2/\|w_\star\|_2)}{(1+\|w_\star\|_2)^2} - \epsilon \tag{A.134}$$

Here φ^2 is the head-error $|(v_\star^\top v_\star)^2 - (v_0^\top v_\star)^2|$. From Corollary A.1, we have with probability at least $1 - \delta$:

$$\cos\theta_k(R, S^{\perp}) \geq \sqrt{\frac{d-n}{d \log \frac{2d}{\delta}}} \tag{A.135}$$

For Gaussian initialization, from Lemma A.7, we have with probability at least $1 - \delta$:

$$\varphi^2 \geq O(\delta(v_\star^\top v_\star)^2) = O(\delta(w_\star^\top w_\star)^2) \tag{A.136}$$

We assumed $\|w_{\star}\|_2$ is a constant, so since big-oh ignores constant factors, we omit the $\|w_{\star}\|_2$ terms. So we get $\varphi^2 = O(\delta)$. Combining everything, we get:

$$\sqrt{L_{\text{ood}}(v_{ft}^t, B_{ft}^t)} \geq O\Big(\frac{\delta}{\sqrt{k}}\sqrt{\frac{d-n}{d\log\frac{2d}{\delta}}}\Big) - \epsilon \tag{A.137}$$

Simplifying the fine-tuning bound: We assumed:

$$\epsilon \leq O\Big(\frac{n(d-n)^{1/2}}{k^{1/2}d^{3/2}}\frac{\delta}{(\log\frac{d}{\delta})^{3/2}}\Big) \tag{A.138}$$

With some algebra, we can show:

$$\Big(\frac{n(d-n)^{1/2}}{k^{1/2}d^{3/2}}\frac{\delta}{(\log\frac{d}{\delta})^{3/2}}\Big) \leq \Big(\frac{\delta}{\sqrt{k}}\sqrt{\frac{d-n}{d\log\frac{2d}{\delta}}}\Big) \tag{A.139}$$

This reduces to showing $n \leq d\log\frac{d}{\delta}$ which is true since we assumed $n \leq d$ in the overparameterized setting. So this means we have:

$$\epsilon \leq O\Big(\frac{\delta}{\sqrt{k}}\sqrt{\frac{d-n}{d\log\frac{2d}{\delta}}}\Big) \tag{A.140}$$

This then gives us that for all small ϵ that we consider, we have:

$$\sqrt{L_{\text{ood}}(v_{ft}^t, B_{ft}^t)} \geq O\Big(\frac{\delta}{\sqrt{k}}\sqrt{\frac{d-n}{d\log\frac{2d}{\delta}}}\Big) \tag{A.141}$$

Linear-probing bound: For linear-probing, from Lemma A.8 we have:

$$\sqrt{L_{\text{ood}}(v_{\text{lp}}^{\infty}, B_0)} \leq O\Big(\frac{d}{n}\log{(\frac{d}{\delta})}\epsilon\|w_{\star}\|_2\Big) \tag{A.142}$$

To simplify presentation, we let $\|w_{\star}\|_2$ be a constant, so we have:

$$\sqrt{L_{\text{ood}}(v_{\text{lp}}^{\infty}, B_0)} \leq O\Big(\frac{d}{n}\log{(\frac{d}{\delta})}\epsilon\Big) \tag{A.143}$$

Take the ratio: We take the ratio of the OOD errors of linear-probing and fine-tuning that we wrote above. Notice that we have an *upper bound* for linear-probing and a *lower bound* for fine-tuning. We get:

$$\frac{\sqrt{L_{\text{ood}}(v_{\text{lp}}^{\infty}, B_0)}}{\sqrt{L_{\text{ood}}(v_{ft}^t, B_{ft}^t)}} \leq O\Big(\frac{\epsilon}{\delta}\sqrt{\frac{kd^3}{n^2(d-n)}(\log\frac{d}{\delta})^3}\Big) \tag{A.144}$$

Finer characterization: $L_{\text{ood}}(v_{\text{lp}}^{\infty}, B_0) < L_{\text{ood}}(v_{ft}^t, B_{ft}^t)$ iff the above ratio is of the losses is less than 1. Simple algebra (moving over terms to the correct side) gives us the desired result, that $L_{\text{ood}}(v_{\text{lp}}^{\infty}, B_0) < L_{\text{ood}}(v_{ft}^t, B_{ft}^t)$ if:

$$\epsilon \leq O\Big(\delta\sqrt{\frac{n^2(d-n)}{kd^3}\frac{1}{(\log\frac{d}{\delta})^3}}\Big) \tag{A.145}$$

Which completes the proof. □

A.4 LP VS FT IN THE GAUSSIAN SETTING (ID)

Restatement of Proposition 3.1. *Suppose $w_{\star} = B_{\star}^{\top}v_{\star} \notin$ rowspace(B_0), and that fine-tuning converges to a local minimum of its loss, then fine-tuning does better ID: $L_{\text{id:subspace}}(v_{\text{ft}}^{\infty}, B_{\text{ft}}^{\infty}) < L_{\text{id:subspace}}(v_{\text{lp}}^{\infty}, B_0)$.*

Proof. **Fine-tuning gets 0 ID loss**: It is well known from prior work (Laurent & von Brecht, 2018) that all local minima are global for optimizing two layer linear networks under convex losses, so if fine-tuning converges to a local minimum, it actually converges to a global minimum of the train loss. Since there exists parameters that achieve 0 loss on the training data (namely, $B_\star, v_\star$), this means fine-tuning gets 0 loss on the training data as well. In noiseless linear regression, 0 loss on the training data implies 0 loss on the span of the training data as well, so:

$$L_{\text{id:subspace}}(v_{\text{ft}}^\infty, B_{\text{ft}}^\infty) = 0 \tag{A.146}$$

Suffices to show linear-probing gets positive ID loss: So it suffices to show that $L_{\text{id:subspace}}(v_{\text{lp}}^\infty, B_0) > 0$. The high level intuition of our proof is that for every subset of k examples of the training data (of size n), there is a unique linear-probing solution that fits these k examples. Since the data is random, the chance that the linear probing solution for one set of k examples agrees with another set of k examples is 0. Since $k < n$, this means that $w_{\text{lp}} = B_0^\top v_{\text{lp}}^\infty$ cannot fit all the training examples, which means it gets positive loss. We formalize this argument below.

Recall that we have $n > k$ training examples. For $i \leq n$, let X_i denote a matrix of shape (i, d) where the rows are the first i training examples. X_i is a random matrix, because the training examples are random.

Linear probing objective on first i examples: Given a head v, consider the linear probing objective on the first i examples:

$$L_i(v) = \|X_i B_0^\top v - X_i B_\star^\top v_\star\|_2^2 \tag{A.147}$$

Unique minimizer of L_i for $i \geq k$: Expanding $L_i(w)$, we see that it is a quadratic form with quadratic matrix $B_0 X_i^\top X_i B_0^\top$. We note that since X_i is a matrix with independent standard Gaussian ($N(0,1)$) entries. B_0 has orthonormal rows, so $B_0 X_i^\top$ is a matrix of shape (k, i) with standard Gaussian entries. So if $i \geq k$, then $B_0 X_i^\top$ has rank k almost surely. This means $B_0 X_i^\top X_i B_0^\top$ has rank k—since it is positive semi-definite and has full rank, it is in fact positive definite. This means that for $i \geq k$, $L_i(v)$ is strongly convex, and has a unique minimum given by:

$$v_i = (B_0 X_i^\top X_i B_0^\top)^{-1} B_0 X_i^\top X_i B_\star^\top v_\star \tag{A.148}$$

This can be shown by noting that $\nabla L_i(v_i) = 0$, and since L_i is strongly convex this implies that v_i is the unique minimizer of L. Since L_i is non-negative with a unique minimizer, any $v \neq v_i$ gives us $L_i(v) > 0$.

v_k **gets the $(k+1)$-th example wrong**: Now, we examine v_k which is the unique solution that minimizes L_k, the loss on the first k examples. The prediction on the $(k+1)$-th example x_{k+1} is: $(B_0^\top v_k)^\top x_{k+1}$. The ground truth target for the $(k+1)$-th example is $(B_\star^\top v_\star)^\top x_{k+1}$. So the difference (error) is:

$$E_{k+1} = [(B_0^\top v_k - B_\star^\top v_\star)^\top x_{k+1}]^2 \tag{A.149}$$

Since $w_\star = B_\star^\top v_\star$ is not in the range of $B_0^\top$ (the rowspace of B_0), we have $B_0^\top v_k - B_\star^\top v_\star \neq 0$. Since x_{k+1} is a unit Gaussian vector, $(B_0^\top v_k - B_\star^\top v_\star)^\top x_{k+1} \neq 0$ almost surely. This means $E_{k+1} > 0$ almost surely.

v_n **must get some example wrong**: Finally, we examine v_n. First we note that the linear-probing loss is a strongly convex quadratic, so gradient flow on all n training examples converges to the solution $w_{\text{lp}} = B_0^\top v_n$. We now consider two cases.

Case 1, $v_n = v_k$: Then $w_{\text{lp}} = B_0^\top v_n$ gets example x_{k+1} wrong, which is in the span of the training data, so $L_{\text{id:subspace}}(v_{\text{lp}}^\infty, B_0) > 0$.

Case 2, $v_n \neq v_k$: Then $L_k(v_n) > 0$ since v_k is the unique minimizer of L_k. This means that the loss of $B_0^\top v_n$ on one of the first k training examples is non-zero, and therefore $L_{\text{id:subspace}}(v_{\text{lp}}^\infty, B_0) > 0$.

Either way, $L_{\text{id:subspace}}(v_{\text{lp}}^\infty, B_0) > 0$ almost surely. Since $L_{\text{id:subspace}}(v_{\text{ft}}^\infty, B_{\text{ft}}^\infty) = 0$, we have $L_{\text{id:subspace}}(v_{\text{ft}}^\infty, B_{\text{ft}}^\infty) < L_{\text{id:subspace}}(v_{\text{lp}}^\infty, B_0)$ which completes the proof. □

A.5 LP-FT

We start by showing a simple proposition, that if the initial feature extractor is perfect, then linear probing recovers the optimal weights.

Proposition A.2. *In the overparameterized linear setting, let $R =$ rowspace(B_0). If $B_0 = B_\star$, and $\cos\theta_k(S, R) > 0$, then $L_{\mathsf{ood}}(v_{\mathsf{lp}}^\infty, B_0) = 0$ for all t.*

Proof. We first show that because $\cos\theta_k(R, S) > 0$, the training loss for linear probing is strongly convex. Recall that the training loss is:

$$\widehat{L}(v, B) = \|XB^\top v - Y\|_2^2 \tag{A.150}$$

Linear probing keeps B fixed as $B_0 = B_\star$ and only tunes v, so we are interested in the Hessian of the loss with respect to v evaluated at $v, B_\star$:

$$\mathrm{Hess}_v\, \widehat{L}(v, B_\star) = 2(B_\star X^\top)(B_\star X^\top)^\top \tag{A.151}$$

For strong convexity, it suffices to show that the min singular value of the Hessian is bounded away from 0 by a constant. Recall the definition of $\cos\theta_k(R, S)$. For some F whose columns form an orthonormal basis for S, we have (since the rows of $B_\star$ form an orthonormal basis for R):

$$\sigma_k(B_\star F) = \cos\theta_k(R, S) > 0 \tag{A.152}$$

Note that $B_\star F$ is a k-by-n matrix, so if the k-th singular value is positive it must be full rank. Since the columns of $X^\top$ span F (since we defined F to be such that the columns of F are an orthonormal basis for S, i.e. the rows of X), this means $B_\star X^\top$ is rank k. But that means the Hessian $(B_\star X^\top)(B_\star X^\top)^\top$ is rank k as well. So the linear probing loss is strongly convex.

Since the loss is strongly convex, there is a unique minimizer, and gradient flow converges to that. However, since we are in the well-specified setting, we know the training loss is:

$$\widehat{L}(v, B_\star) = \|XB_\star^\top v - XB_\star^\top v_\star\|_2^2 \tag{A.153}$$

So $v = v_\star$ achieves 0 loss and must be the (unique) minimizer. Therefore we have shown that linear probing converges to the unique minimizer $v_{\mathsf{lp}}^\infty = v_\star$, which attains 0 loss, as desired.

Note that the entire proof works out if $B_0 = UB_\star$ for some rotation matrix U. In that case, the Hessian becomes $2U(B_\star X^\top)(B_\star X^\top)^\top U^\top$ which is still rank k, since multiplying by square rotation matrices does not change the rank. In this case, the minimizer of the loss is $v = Uv_\star$, since $(UB_\star)^\top(Uv_\star) = B_\star^\top v_\star$. So linear probing converges to $v_{\mathsf{lp}}^\infty = Uv_\star$, which achieves 0 loss, as desired. □

Restatement of Proposition 3.2. *Suppose we have perfect pretrained features $B_0 = UB_\star$ for some rotation U. Let $R =$ rowspace(B_0). Under the non-degeneracy conditions $\cos\theta_k(R, S) \neq 0, \cos\theta_k(R, S^\perp) \neq 0$:*

$$\forall t, L_{\mathsf{ood}}(B_{\mathsf{ft}}(t)^\top v_{\mathsf{ft}}(t)) > 0, \text{ if } v_0 \sim \mathcal{N}(0, \sigma^2 I) \text{ is randomly initialized (FT)} \tag{A.154}$$

$$\forall t, L_{\mathsf{ood}}(B_{\mathsf{ft}}(t)^\top v_{\mathsf{ft}}(t)) = 0, \text{ if } v_0 \text{ is initialized to } v_{\mathsf{lp}}^\infty \text{ (LP-FT)} \tag{A.155}$$

Proof. We first use Proposition A.2, which in the proof we showed still works if $B_0 = UB_\star$ for some rotation matrix U (which doesn't have to be identity). We get that $v_{\mathsf{lp}}^\infty = Uv_\star$. Then we have $B_0^\top v_{\mathsf{lp}}^\infty = B_\star^\top v_\star = w_\star$.

We now just show that the gradients with respect to the training loss $\widehat{L}$ at $(v_{\mathsf{lp}}^\infty, B_0)$ is 0, so gradient flow does not update the parameters at all.

The training loss is:

$$\widehat{L}(v, B) = \|XB^\top v - XB_\star^\top v_\star\|_2^2 \tag{A.156}$$

The derivative with respect to v is:

$$\partial_v \widehat{L}(v, B) = 2BX^\top(XB^\top v - XB_\star^\top v_\star) \tag{A.157}$$

Table 3: **OOD accuracies** with 90% confidence intervals over 3 runs, for each of the three OOD domains in the split of DomainNet used by Tan et al. (2020); Prabhu et al. (2021). LP does better than FT across the board, and LP-FT does the best.

	Real	Painting	Clipart
Fine-tuning	55.29 (0.52)	50.26 (0.98)	60.93 (2.15)
Linear probing	**87.16 (0.18)**	74.50 (0.58)	77.29 (0.12)
LP-FT	**86.82 (0.51)**	**75.91 (0.73)**	**79.48 (0.90)**

Then since $B_0^\top v_{\mathsf{lp}}^\infty = B_\star^\top v_\star$, we have:

$$\partial_v \widehat{L}(v_{\mathsf{lp}}^\infty, B_0) = 0 \tag{A.158}$$

Next, the derivative with respect to B is:

$$\partial_B \widehat{L}(v, B) = 2v(XB^\top v - XB_\star^\top v_\star)^\top X \tag{A.159}$$

Then since $B_0^\top v_{\mathsf{lp}}^\infty = B_\star^\top v_\star$, we have:

$$\partial_B \widehat{L}(v_{\mathsf{lp}}^\infty, B_0) = 0 \tag{A.160}$$

So since both the derivatives are 0, we have $\partial_t v_{\mathsf{ft}}(t) = 0$ and $\partial_B B_{\mathsf{ft}}(t) = 0$, which means the parameters don't change at all—at all times t we have $v_{\mathsf{ft}}(t) = Uv_\star$ and $B_{\mathsf{ft}}(t) = UB_\star$ which gives us zero OOD loss: $L_{\mathsf{ood}}(B_{\mathsf{ft}}(t)^\top v_{\mathsf{ft}}(t)) = 0$ as desired. □

B MORE INFORMATION ON EXPERIMENTS

In this Appendix, we include more details on the datasets, pretraining methods, and adaptation methods. We also include the OOD accuracies for fine-tuning and linear-probing if we early stop and choose the learning rate based on OOD data, where we see that linear-probing is still typically better than fine-tuning OOD. Finally, we include results for additional baselines, pretraining models, larger scale datasets, and conclude with a discussion about the effective robustness of LP-FT.

B.1 DATASET AND METHOD DETAILS

We use a diverse range of datasets and pretraining strategies.

- **CIFAR-10 → STL**: We fine-tune or linear probe on CIFAR-10 (Krizhevsky, 2009) and test on STL (Coates et al., 2011). This is a benchmark used in domain adaptation papers (French et al., 2018). CIFAR-10 and STL share 9 classes, so we follow the common practice of omitting the unshared class in STL (which is the 'monkey' class) when reporting accuracies. We use a publicly available MoCo-v2 ResNet-50 checkpoint pretrained on unlabeled examples from ImageNet-1k (Russakovsky et al., 2015), and fine-tune for 20 epochs.

- **DomainNet**: We use the dataset splits in Peng et al. (2019) which is also used by follow-up work, e.g., in Prabhu et al. (2021). We use the 'sketch' domain as ID, and all other domains ('real', 'clipart', 'painting') as OOD, and in the main paper we report the average accuracies across the OOD domains. In Table 3 we see that the *same trends hold for each of the three OOD domains*. We use a CLIP (Radford et al., 2021) pretrained ResNet-50 model, and fine-tune for 50 epochs (since this is a smaller dataset).

- **Living-17** and **Entity-30**: We use a publicly available MoCo-v2 ResNet-50 checkpoint pretrained on unlabeled examples from ImageNet-1k (Russakovsky et al., 2015), and fine-tune for 20 epochs. Note that Living-17 and Entity-30 are subpopulation shifts derived from ImageNet, but the pretraining is done on unlabeled data and does not see any OOD labels, following the pretraining and fine-tuning strategy in Cai et al. (2021). Entity-30 is a relatively large dataset that contains around 140K training examples.

Table 4: **OOD accuracies** with 90% confidence intervals over 3 runs, when fine-tuning gets to choose learning rate and early stop, and linear probing gets to choose ℓ_2 regularization weights, on OOD data. We see that linear probing still typically does better OOD (the only flip from before is on FMoW).

	CIFAR-10.1	STL	Ent-30	Liv-17	DomNet	FMoW
FT	**92.27 (0.36)**	85.97 (0.38)	64.09 (0.19)	78.63 (0.53)	59.43 (2.49)	**40.23 (3.12)**
LP	82.67 (0.22)	**86.53 (0.01)**	**69.15 (0.13)**	**82.39 (0.14)**	**79.91 (0.24)**	37.12 (0.01)

Table 5: **In-distribution (ID)**: Average distance that features move before and after fine-tuning or LP-FT, multiplied by 100 to make things easier to read. For linear probing the numbers are all 0, since the features are not tuned. As predicted by our theory, we see that features for ID examples (this table) move more than features for OOD examples (Table 8). Both sets of features change substantially less for LP-FT. As usual we show 90% confidence intervals over three runs.

	CIFAR-10	Entity-30	Living-17	DomainNet	FMoW
FT	2.23 (0.03)	3.05 (0.02)	1.88 (0.01)	207.6 (12.31)	4.87 (0.15)
LP-FT	0.07 (0.00)	0.03 (0.01)	0.11 (0.01)	0.19 (0.03)	0.57 (0.19)

- **FMoW Geo-shift**: We adapt the version of the dataset from (Koh et al., 2021). We use training data from 'North America' to fine-tune or linear probe, and then evaluate on validation data from Africa and Europe. We use a MoCo-TP (Ayush et al., 2020) checkpoint, pretrained on unlabeled FMoW satellite images. We fine-tune for 50 epochs here since the ID training dataset is smaller (around 20K examples).

- **CIFAR-10 → CIFAR-10.1** (Recht et al., 2018): We follow the same protocols as CIFAR-10 → STL, except we test on CIFAR-10.1.

We note that our results say that the pretraining has to be good (e.g., at least get reasonable accuracy ID) for linear probing to outperform fine-tuning OOD. So, for example, we use a model pretrained on unlabeled satellite images for the satellite image dataset—if we pretrain the model on ImageNet, we expect that fine-tuning might do better. Similarly, for DomainNet we use a CLIP pre-trained model, which is pretrained on the very large WebImageText dataset, and sees a variety of photo and sketch like images. Pretraining on ImageNet alone does not lead to high accuracies on DomainNet (features are not very good), so we do not necessarily expect linear probing to outperform fine-tuning with these lower quality features (for example, see the MoCo ablation in our main paper where we used a worse pretrained model, and fine-tuning did better OOD).

As a sanity check of our implementation, fine-tuning did substantially better than training from scratch on all datasets (both ID and OOD) and matched existing fine-tuning numbers where available (e.g. ResNet50 on CIFAR-10 (Chen et al., 2020b) and Entity-30 (Cai et al., 2021)).

Fine-tuning and linear probing both do substantially better than training from scratch, ID and OOD, across the datasets. For example, on Living-17, training from scratch gets 89.3% ID and 58.2% OOD which is over 5% worse ID and nearly 20% worse OOD, than all the adaptation methods. For reference linear probing gets 96.5% ID and 82.2% OOD, and fine-tuning gets 97.1% ID and 77.8% OOD. This is even though training from scratch was run for 300 epochs, which is 15 times longer than fine-tuning and LP-FT.

B.2 Target early stopping

In the main paper, one ablation we mention is early stopping each fine-tuning method and choose the best learning rate based on target validation accuracy. As expected, fine-tuning does improve a little, but linear probing (average accuracy: 73.0%) is still better than fine-tuning (average accuracy: 70.1%). Table 4 shows the full results.

Table 6: **Out-of-distribution (OOD)**: Average distance that features move before and after fine-tuning or LP-FT, multiplied by 100 to make things easier to read. For linear probing the numbers are all 0, since the features are not tuned. As predicted by our theory, we see that features for ID examples (Table 5) move more than features for OOD examples (this table). Both sets of features change substantially less for LP-FT. As usual we show 90% confidence intervals over three runs.

	STL	Entity-30	Living-17	DomainNet	FMoW
FT	1.70 (0.04)	2.60 (0.02)	1.67 (0.01)	159.97 (16.23)	5.62 (0.30)
LP-FT	0.04 (0.00)	0.02 (0.00)	0.09 (0.01)	0.18 (0.02)	0.54 (0.17)

B.3 Feature change

We examine how much the features changed for ID and OOD examples in each dataset. Specifically, for each dataset, for each input example in the held out validation set, we computed the Euclidean distance of the ResNet-50 features before and after fine-tuning. We averaged these numbers across the dataset, showing the results for ID validation examples in Table 5, and for OOD examples in Table 8.

The feature distortion theory predicts that the features for ID examples change more than for OOD examples. This bears out in 9 out of 10 cases, that is all cases except for FT on FMoW. To see this, compare each cell in Table 5 with the corresponding cell in Table 8—the former is higher in 9 out of 10 cases.

The feature distortion theory says that this large feature change is caused because the head is randomly initialized—since the head needs to be updated by a large amount, the feature extractor is also updated a lot because the updates are coupled. Our theory predicts that if the head is initialized via linear probing then the feature extractor should change a lot less for both ID and OOD examples. As predicted by the theory, across all the datasets in Table 5 and Table 8, the features change a lot less for LP-FT than for FT. For example, on CIFAR-10, the features change 30× less for LP-FT than for FT.

These results suggest that fine-tuning underperforms OOD, and LP-FT does well ID and OOD, for the reasons predicted by the feature distortion theory.

B.4 Additional architectures, fine-tuning methods

The main contributions of our paper are conceptual understanding and theory. However, to strengthen the empirical investigation we ran two additional models (a CLIP vision transformer and CLIP ResNet-50), as well as three additional fine-tuning heuristics. We focus on the Living-17 dataset because some of these ablations require lots of compute and can take a long time to run on all the datasets.

Architectures and pretraining source: In the main paper, we showed results when initializing with a MoCo-v2 ResNet-50 model pretrained on unlabeled ImageNet examples. Here we examine how the results change when we 1. Use a ResNet-50 model pretrained on CLIP's WebImageText dataset, and, 2. Use a much larger vision transformer model (ViT-B/16) pretrained on CLIP's WebImageText dataset—this is the largest publicly available CLIP model at the time of writing. We see that similar findings to our main paper hold—fine-tuning does better than linear probing ID, but does worse than linear probing ('underperforms') OOD. Finally, LP-FT does better than both methods ID, and closes most (75%-90%) of the gap OOD.

These results are from early stopping on ID validation data. If we early stop on OOD validation data, LP-FT achieves $87.9 \pm 0.4\%$ OOD accuracy, and LP gets $88.3 \pm 0.2\%$ OOD accuracy and here there is no statistically significant difference between the two. On the other hand, even if we early stop on OOD validation data, fine-tuning gets $84.4 \pm 0.5\%$ OOD accuracy which is lower.

Fine-tuning heuristics: Transfer learning (initializing with a pretrained model, and then adapting it to a downstream task) is the standard way to build modern ML models, because it improves accuracy and speeds up training. Since this paradigm is so widely used, there are many heuristics people use when training their models (as mentioned in the main paper, LP-FT has sometimes been used as a

Table 7: ID and OOD accuracies on Living-17 using a CLIP ResNet-50 model pretrained on the WebImageText dataset, instead of unlabeled ImageNet examples. Similar findings hold—here fine-tuning does similarly to linear probing ID, but does worse than linear probing OOD. LP-FT does better than both ID, and closes 75% of the gap OOD. As usual we show 90% confidence intervals over three runs.

	ID	OOD
LP	94.7 (0.2)	**78.6 (0.5)**
FT	94.7 (0.1)	67.3 (0.8)
LP-FT	**95.6 (0.2)**	77.0 (0.6)

Table 8: ID and OOD accuracies on Living-17 using a CLIP ViT-B/16 (Vision Transformer) model pretrained on the WebImageText dataset, instead of unlabeled ImageNet examples. This is the largest publicly available CLIP model that we could find. The same findings hold—fine-tuning does better than linear probing ID, but does worse than linear probing OOD. LP-FT does better than both ID, and closes 86% of the gap OOD. As usual we show 90% confidence intervals over three runs.

	ID	OOD
LP	97.5 (0.1)	**87.6 (0.5)**
FT	97.8 (0.0)	81.5 (2.1)
LP-FT	**98.0 (0.0)**	86.1 (0.1)

heuristic as well, although not in the context of OOD). We showed that LP-FT is one way to do well ID and OOD, but we hope that our theory leads to even better fine-tuning algorithms.

In this section, we compare LP-FT with additional fine-tuning heuristics: using a larger learning rate for the head layer, regularizing the features towards their original values, and side-tuning (Zhang et al., 2020) where we freeze the features but add a side-network.

The intuitions from our theory suggest two other potential ways to improve OOD accuracy: 1. We could use a higher learning rate on the linear layer, so that the linear layer learns quicker and the features do not get as distorted, and 2. We could regularize the weights of the feature extractor towards the pretrained initialization, to prevent feature distortion. These heuristics have been used in prior work on fine-tuning as well, for example method 2 corresponds to L2-SP in (Li et al., 2018).

We run these two approaches on Living-17. For approach (1), we use a 10× higher learning rate for the linear layer, and for approach (2) we regularize the Euclidean distance between the current feature extractor weights (so ignoring the linear head) from the pretrained weights, multiplying by a hyperparameter λ. We grid search over the same learning rates as fine-tuning for both methods, and in addition for (2) we grid search over $\lambda \in \{1.0, 0.1, 0.01, 0.001, 0.0001\}$, so this amounts to sweeping over 30 hyperparameters as opposed to just 6 for fine-tuning and LP-FT. For each hyperparameter configuration we run 3 replication runs with different seeds to reduce the estimation variance, and early stop and model select using ID data just like for fine-tuning and LP-FT. Just like for fine-tuning and LP-FT, we use a cosine learning rate decay and train for the same number of epochs. Indeed, we find that both (1) and (2) are able to close part of the OOD gap between fine-tuning and linear-probing. However, LP-FT does better than both methods ID and OOD. The full results are in Table 9.

We also compare with another method, (3) side-tuning (Zhang et al., 2020). Side-tuning freezes the pretrained features $g(x)$ but trains another 'side' model $s(x)$, and then outputs $v^\top(g(x) + h(x))$, where the head v and the parameters of the side model s are tuned. The intuition for trying this is that side-tuning also preserves the pretrained features which likely reduces feature distortion. In the supplementary of Zhang et al. (2020) they use a ResNet-50 for both the original model and the side model in their vision experiments, so we do the same. We sweep over twelve learning rates $(3 \cdot 10^{-5}, 1 \cdot 10^{-4}, 3 \cdot 10^{-4}, \ldots, 1.0, 3.0, 10.0)$, with three replication runs with different seeds for each learning rate. Just like for fine-tuning and LP-FT, we use a cosine learning rate decay and train for the same number of epochs, and we early stop and model select using ID validation data. We

Table 9: ID and OOD accuracies on Living-17 including three additional fine-tuning heuristics, where we (1) Use a 10× larger learning rate for the head, or (2) Regularize the Euclidean distance of the feature extractor weights to the pretrained initialization, and (3) side-tuning where we freeze the pretrained model but add a side network that is fine-tuned. As a sanity check, all methods do better than training from scratch ID and OOD, and we show 90% confidence intervals over three runs. As per the intuitions from the feature distortion theory, these methods do mitigate feature distortion to some extent and improve OOD accuracy over fine-tuning. LP-FT does better than all methods ID and OOD—nonetheless, we believe that LP-FT is just the first step and hope that our theory can be used to inspire or derive better algorithms.

	ID	OOD
Scratch	92.4 (1.3)	58.2 (2.4)
LP	96.5 (0.1)	82.2 (0.2)
FT	97.1 (0.1)	77.7 (0.7)
FT (10x Linear)	97.2 (0.2)	80.4 (0.3)
FT (regularized)	97.1 (0.2)	80.0 (0.4)
Side-tuning	95.5 (0.4)	81.0 (0.7)
LP-FT	**97.8 (0.1)**	**82.6 (0.3)**

checked that the best learning rate was not at the boundary of the grid search. On OOD, side-tuning (81.0%) improves over fine-tuning (77.7%). However, side-tuning doesn't do as well ID. LP-FT did better ID and OOD. This could be because side-tuning does not get to refine the pretrained features for the ID task—while the side-network is powerful enough to learn good features, it is initialized randomly and effectively trained from scratch, so it might not be able to learn these good features on the limited sized training dataset (around 40K examples). The results are also in Table 9.

We also include results for training from scratch in Table 9—these results are from Santurkar et al. (2020). Note that training from scratch was done for 450 epochs, whereas fine-tuning was done for 20 epochs. As a sanity check, all the fine-tuning methods and linear probing do substantially better than training from scratch, both ID and OOD.

B.5 Larger scale datasets

In our original paper we used popular distribution shift datasets like DomainNet, Breeds, CIFAR-10 → STL and the satellite remote sensing dataset FMoW. Here, we see that our findings hold up in larger scale datasets. Specifically, we linear probe or fine-tune on ImageNet (Russakovsky et al., 2015), and evaluate on ImageNetV2 (Recht et al., 2019), ImageNet-R (Hendrycks et al., 2020), ImageNet-A (Hendrycks et al., 2019b), and ImageNet-Sketch (Wang et al., 2019). We begin with a CLIP ViT-B/16 (vision transformer), the largest publicly available CLIP model (Radford et al., 2021), which is much larger than a ResNet-50. This model is pretrained on the WebImageText dataset (Radford et al., 2021).

Recall that our theory says that linear probing does better than fine-tuning when the shift is large. We expect linear probing to do better on ImageNet-R, ImageNet-A, ImageNet-Sketch. We expect fine-tuning to do better on ImageNetV2 because it is collected by "repeating the dataset curation process" (Recht et al., 2019) of ImageNet—this is similar to CIFAR-10 → CIFAR-10.1 in our original paper.

Training details: since the dataset is large (over one million training images), we did not run replication runs. However, we swept over three learning rates for fine-tuning (0.0001, 0.0003, 0.001) and linear probing (0.01, 0.03, 0.1)—as is standard we use larger learning rates for linear probing. Both fine-tuning and linear probing were run for 10 epochs over the ImageNet training data, with batch size 128, and cosine learning rate decay. As in our other experiments, we early stopped the models and selected the best learning rate to maximize in-distribution validation accuracy (on ImageNet). However, we did not find the results to be sensitive to hyperparameter choices.

We find that fine-tuning gets 2% higher accuracy ID than linear probing, but averaged across the four OOD datasets fine-tuning gets 10% lower accuracy OOD. The ID results are in Table 10 and the OOD results are in Table 11. As expected, fine-tuning does better than linear probing ID (ImageNet validation) and on ImageNetV2 which replicates the collection process of ImageNet and is a small

Table 10: **ID** accuracy of a CLIP ViT-B/16 model fine-tuned on ImageNet training data. As expected, we see that fine-tuning does better than linear probing in-distribution.

	ImageNet
Fine-tuning	**81.7**
Linear probing	79.7

Table 11: **OOD** accuracies of a CLIP ViT-B/16 model fine-tuned on ImageNet training data, on four standard OOD datasets. Fine-tuning does worse than linear probing when the distribution shift is large. The only case where fine-tuning does better is ImageNetV2, which is collected by "repeating the dataset curation process" of ImageNet. This suggests that our findings hold up in larger scale settings as well.

	ImageNetV2 (small shift)	Renditions	Sketch	ImageNet-A
Fine-tuning	**71.5**	52.4	40.5	27.8
Linear probing	69.7	**70.6**	**46.4**	**45.7**

shift. However, fine-tuning does worse than linear probing OOD on ImageNet-R, ImageNet-A, and ImageNet-Sketch.

B.6 Discussion of effective robustness

LP-FT gets higher OOD accuracy than fine-tuning, but it sometimes gets higher ID accuracy as well. Taori et al. (2020) and Miller et al. (2021) show that OOD accuracy can often be correlated with ID accuracy, and suggest examining the effective robustness: intuitively the extra gain in OOD accuracy than can be predicted from improved ID accuracy alone. Is LP-FT simply better in-distribution, or does it have higher effective robustness as well?

We start out by noting that linear probing clearly has higher effective robustness in most of our datasets. Linear probing does worse than fine-tuning ID so based on the effective robustness framework we would expect it to do worse than fine-tuning OOD as well. However, linear probing does better than fine-tuning OOD and therefore has effective robustness.

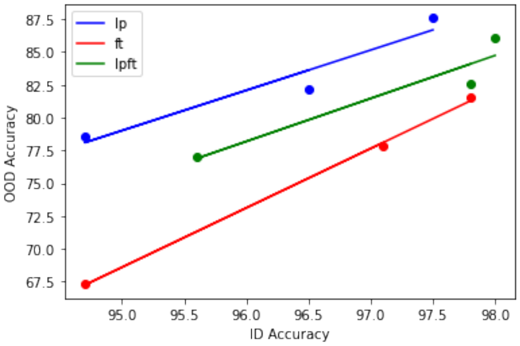


Figure 3: We plot the OOD accuracy against ID accuracy on Living-17 for the three methods we consider, when we start from three different pretrained models (CLIP ResNet-50, CLIP ViT-B/16, MoCo-V2 ResNet-50). The line for linear probing and LP-FT lie above fine-tuning which suggests that they have higher effective robustness. Each point is produced by averaging over three random seeds.

The solutions found by LP-FT also appear to have higher effective robustness than fine-tuning, because when they have similar ID accuracy, LP-FT does much better OOD. For a few pieces of evidence:

1. On CIFAR-10 → STL, there is no statistically significant difference between FT and LP-FT on ID, but LP-FT gets 8% higher accuracy OOD in Table 2.

2. If we look at checkpoints earlier in training for CIFAR-10 → STL we can exactly equalize ID accuracy and compare OOD accuracies. In-distribution, LP-FT and FT both get 97.2% accuracy, but OOD, LP-FT (90.2%) is much better than FT (81.8%).

3. Finally, in Figure 3 we plot the OOD accuracy against the ID accuracy for fine-tuning and LP-FT on Living-17. We plot these for three different pretrained models (CLIP ResNet-50, CLIP ViT-B/16, MoCo-V2 ResNet-50). We see that the ID-OOD line for LP-FT is above the line for FT indicating effective robustness.

Note that higher effective robustness does not mean a method is better. For example, a method A can have higher effective robustness B by doing a lot worse in-distribution even when they have the same OOD accuracy. In this case, A is clearly inferior since it does worse ID and same OOD, but has higher effective robustness because of its worse ID accuracy.

We believe the finding that linear probing and LP-FT has higher effective robustness than fine-tuning when the distributon shift is large is particularly interesting because Taori et al. (2020) and Miller et al. (2021) show that it is uncommon for methods to have higher effective robustness. In our case linear probing and LP-FT appear to consistently have higher effective robustness which suggests that with proper transfer learning methods we can get both high in-distribution accuracy and higher effective robustness.

C Additional related work

Theoretical analysis of overparameterized models. Modern deep learning presents an interesting paradigm for theoretical analysis where the number of parameters is much larger than the number of training points. The model class is highly expressive and several solutions obtain zero training loss even in the presence of noise. Such overparameterized models have received a lot of interest recently especially with a focus on understanding "benign overfitting" or the phenomenon where fitting noisy training data to zero loss leads to classifiers that generalize well. By analyzing different linear overparameterized settings Belkin et al. (2019); Hastie et al. (2019); Bartlett et al. (2019); Muthukumar et al. (2020); Mei & Montanari (2019); Bibas et al. (2019) study various statistical properties such as the "double descent curve" in addition to benign overfitting. One important aspect of overparameterized models is that there is no unique minimizer of the training loss. We need some *inductive bias* which is typically implicit via the optimization procedure. Prior works study the statistical properties of the explicit inductive bias of minimum norm interpolation. In contrast, we study the effect of gradient based optimization from a particular pretrained initialization where we effectively capture the exact implicit inductive bias of gradient based fine tuning.