Model Stock: All we need is just a few fine-tuned models
– Appendix –

Dong-Hwan Jang[1,2⋆] Sangdoo Yun[1†] Dongyoon Han[1†]

[1] NAVER AI Lab [2] Samsung Advanced Institute of Technology (SAIT)
[†] corresponding authors

In this Appendix, we provide in-depth analysis and additional insights to complement the main text of our study on Model Stock, our novel approach to fine-tuning and weight merging. The contents are summarized as follows:

- We examine the angle norm consistency of fine-tuned weights across various settings in §A, extending the observations discussed in §2.1.
- We provide detailed proofs of geometric properties of fine-tuned weights in §B.
- We study the importance of reducing variance for performance in out-of-distribution scenarios in §C, showcasing the test error landscape across various datasets and elaborating on the explanations in §2.2.
- We provide detailed proofs in §D for the optimal interpolation ratio in our method §3.
- We discuss prior studies through the lens of our findings in §E.
- We provide an additional analysis of the interpolation ratio in §F.
- We present experimental settings of §4 in §G.
- We present additional experiments of Model Stock in §H

Each section aims to offer a comprehensive understanding of our method's underlying principles and its broad applicability in machine learning.

A Angle and Norm Consistency

We argue that, as discussed in §2.1, angles and norms of fine-tuned weights would remain consistent across fine-tuned models, independent of various factors. These factors include architecture type (ViTs [4], ResNet [7], ConvNeXt [14]), optimizers (SGD, AdamW [15]), augmentations (RRC [20], RandAug [3]), datasets (CIFAR [9], ImageNet [19]), or the initialization of the classifier (zero-shot, LP as in LP-FT [10]). We depict the layer-wise angle and norm of 5 fine-tuned weights for each category based on different random seeds. We give detailed illustrations for each setting at the end of the Appendix to enhance readability (refer to Fig. H–O). Across all these settings, the angle and norm of weights exhibit a surprising level of consistency.

⋆ Work done during an internship at NAVER AI Lab.

A.1 Analysis on layer-wise tendency

The layer-wise angle and norm across various settings are shown in Fig. H–L. We visualize with every weight of attentions/convolutions (Attention/Conv), multi-layer perceptrons (MLP), normalizations (LayerNorm and BatchNorm), a classifier (Classifier), individual bias (Bias), and the remaining layers (*i.e.*, the patchification layer, positional embedding, class embedding, and projection layer). We further display *All* in each figure, which denotes the concatenation of the weights of entire layers.

The layer-wise analysis reveals an interesting trend: Bias and classifier layers demonstrate smaller angles than attention and MLP layers. In other words, bias and classifier layers exhibit lower randomness and more reliable updates than attention and MLP layers. It is important to note that as the angle decreases, the pre-trained model is less utilized for merging (refer to Eq. (2)). This indicates that bias and classifier layers focus more on fine-tuned models and rely less on the pre-trained model, whereas attention and MLP layers depend less on the fine-tuned model (*i.e.*, $t_{\text{bias}}, t_{\text{clf}} > t_{\text{attn}}, t_{\text{mlp}}$). This observation extends the findings of previous works such as BitFit [26] and LP-FT [10]. In the case of BitFit and LP (*i.e.*, the first step of LP-FT), bias and classifier layers fully utilize fine-tuning, while other layers (attention and MLP) rely on pre-trained models.

These traits could offer new insights into parameter-efficient transfer learning (PETL) [6,8,13,26] and layer-wise fine-tuning [10,11,21,22]. Maintaining weights with high randomness (higher angles) while updating on biases and classifier weights with lower randomness and fewer parameters would be an efficient fine-tuning strategy. PETL has been exploring this direction but has not yet provided solid reasons why certain layers are more effective than others. Our analysis suggests that one reason could be the lower randomness (or variance) of these layers, as indicated by the angle trend per layer.

A.2 Maintaining consistency during training

We further argue that the consistency we observed is maintained while training progresses, as illustrated by multiple thin shells in Fig. 5. To demonstrate that the angle and norm of fine-tuned models remain consistent during the entire training process, we plot their relationship across weights for every epoch in Fig. M. Please note that the angle is consistent across differently seeded models at the same timestamp (*i.e.*, $\mathbf{w}_1|_{t=t_1}$ and $\mathbf{w}_2|_{t=t_1}$), not across models at different timestamps (*i.e.*, $\mathbf{w}_1|_{t=t_1}$ and $\mathbf{w}_1|_{t=t_2}$). The observed trend is as follows: as training progresses, the angle between weights steadily decreases. This analysis uses the CLIP ViT-B/32 model fine-tuned on ImageNet-1K with five random seeds.

A.3 Filter-wise analysis of weights

Li *et al.* [12] showed that when evaluating the robustness of a neural network by adding random noise to certain weights, performance analysis based on adding

filter-wise noise (*i.e.*, adding noise for each row in all weight matrices) aligns more closely with the generalization performance than adding layer-wise noise does. Inspired by this observation, we investigate the possibility that the weight distribution may follow a filter-wise Gaussian distribution and adapt this concept to our method (see the performance analysis in §H.5). Fig. N illustrates the angle distribution filter-wise. The angle exhibits much larger standard deviations than the layer-wise distribution. This could be attributed to the reduction in dimensionality. As the number of dimensions decreases, it becomes challenging to approximate the norm as a constant value.

A.4 Analysis on non-CLIP models

To verify if this key observation also applies to non-CLIP models, we analyze the geometric patterns of fine-tuned weights trained using the DeiT [23] method (*i.e.*, pre-trained on ImageNet-21K). Fig. O displays the angle and norm of 10 DeiT-base models first pre-trained on ImageNet-21K [19] and then fine-tuned on ImageNet-1K. We find that weights pre-trained with ImageNet-21K also exhibit *consistent angle and norm*, indicating that our observation may be valid beyond CLIP fine-tuning scenarios as well.

B Detailed Proof for Geometric Properties of Fine-tuned Weights

For all indices i, j within the set $[1, N]$, where N denotes the sufficiently large number of fine-tuned weights, we derive one lemma and three propositions based on the foundational observation described in Eq. (1):

Lemma: $\mathbf{w}_i \cdot \boldsymbol{\mu} = \boldsymbol{\mu} \cdot \boldsymbol{\mu} = l^2 \cos\theta$.

Proof:

$$\begin{aligned}\mathbf{w}_i \cdot \boldsymbol{\mu} &= \lim_{N\to\infty} \frac{1}{N}\mathbf{w}_i \cdot \sum_{k=1}^{N} \mathbf{w}_k = \lim_{N\to\infty} \frac{1}{N}(l^2 + (N-1) * l^2 \cos\theta) \\ &= l^2 \cos\theta.\end{aligned}$$

Similarly,

$$\begin{aligned}\boldsymbol{\mu} \cdot \boldsymbol{\mu} &= \lim_{N\to\infty} \frac{1}{N^2} \sum_{k=1}^{N} \mathbf{w}_k \cdot \sum_{l=1}^{N} \mathbf{w}_l = \lim_{N\to\infty} \frac{1}{N^2}(N * l^2 + N(N-1) * l^2 \cos\theta) \\ &= l^2 \cos\theta. \quad \square\end{aligned}$$

Proposition 1: $\|\mathbf{w}_i - \boldsymbol{\mu}\| = \text{constant}$.

Proof:

$$\begin{aligned}\|\mathbf{w}_i - \boldsymbol{\mu}\|^2 &= (\mathbf{w}_i - \boldsymbol{\mu}) \cdot (\mathbf{w}_i - \boldsymbol{\mu}) \\ &= \mathbf{w}_i \cdot \mathbf{w}_i - 2\mathbf{w}_i \cdot \boldsymbol{\mu} + \boldsymbol{\mu} \cdot \boldsymbol{\mu} \\ &= l^2 - 2l^2 \cos\theta + l^2 \cos\theta \quad \text{(by Lemma)} \\ &= l^2(1 - \cos\theta) \quad \text{(constant)} \quad \square\end{aligned}$$

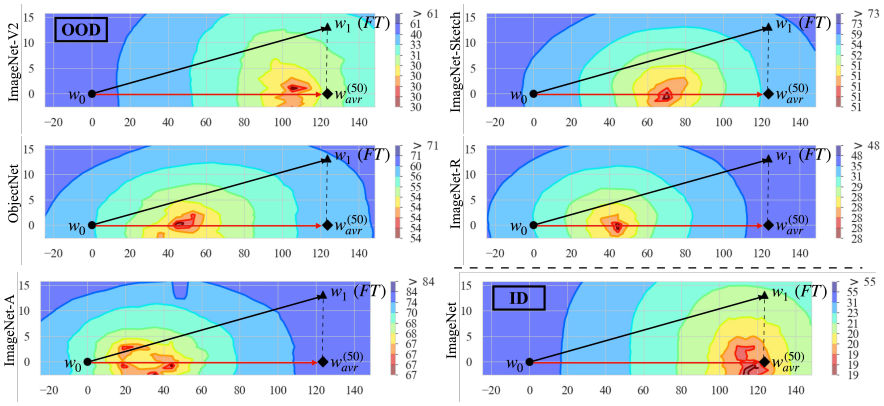


Fig. A: Test error landscape on OOD datasets. We depict the test error landscape on ImageNet-V2, -Sketch, ObjectNet, ImageNet-R, and -A (from left to right, from top to bottom, respectively) on the plane containing pre-trained model ($\mathbf{w}_0$), fine-tuned model ($\mathbf{w}_1$), and the pseudo-center of fine-tuned weights ($\mathbf{w}_{\text{avr}}^{(50)}$). The local optima for the OOD datasets always lie on the line segment $\overline{\mathbf{w}_0\mathbf{w}_{\text{avr}}^{(50)}}$.

Proposition 2: $(\mathbf{w}_0 - \boldsymbol{\mu}) \perp (\mathbf{w}_i - \boldsymbol{\mu})$.
Proof:

$$\begin{aligned}(\mathbf{w}_0 - \boldsymbol{\mu}) \cdot (\mathbf{w}_i - \boldsymbol{\mu}) &= -\boldsymbol{\mu} \cdot (\mathbf{w}_i - \boldsymbol{\mu}) \\ &= 0 \quad \text{(by Lemma)} \quad \square\end{aligned}$$

Proposition 3: $(\mathbf{w}_i - \boldsymbol{\mu}) \perp (\mathbf{w}_j - \boldsymbol{\mu})$.
Proof:

$$\begin{aligned}(\mathbf{w}_i - \boldsymbol{\mu}) \cdot (\mathbf{w}_j - \boldsymbol{\mu}) &= \mathbf{w}_i \cdot \mathbf{w}_j - \mathbf{w}_i \cdot \boldsymbol{\mu} - \mathbf{w}_j \cdot \boldsymbol{\mu} + \boldsymbol{\mu} \cdot \boldsymbol{\mu} \\ &= 0 \quad \text{(by Eq. (1) \& Lemma)} \quad \square\end{aligned}$$

C Importance of Reducing Weight Variance on Performance under Distribution Shifts

In demonstrating the significance of variance reduction for robustness in out-of-distribution (OOD) scenarios, we analyze the test error landscape as in §2.2. As shown in Fig. A, we examine the error landscape across various OOD datasets, including ImageNet-V2, ImageNet-Sketch, ObjectNet, ImageNet-R, and ImageNet-A (from top to bottom). This landscape is plotted on a plane defined by the weights of a pre-trained model ($\mathbf{w}_0$), a fine-tuned model ($\mathbf{w}_1$), and the center of the fine-tuned weights, which is approximated by averaging 50 fine-tuned weights ($\mathbf{w}_{\text{avr}}^{(50)}$). A notable pattern emerges where the local optima for these datasets consistently align with the line segment connecting $\mathbf{w}_0$ and $\mathbf{w}_{\text{avr}}^{(50)}$.

Though the exact location of local minima differs depending on the dataset type, it has a common point that the minima are aligned on the line between the

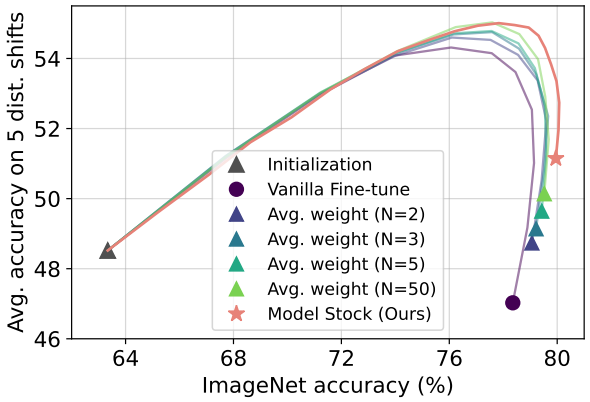


Fig. B: ID vs. OOD accuracy along WiSE-FT [25] curves for averaged models. As the number of weights used for averaging increases, the corresponding WiSE-FT curves demonstrate improvements in the ID-OOD trade-off.

weight center and pre-trained model rather than the line between the fine-tuned weight and pre-trained model. Consequently, not only does the averaged weight exhibit higher performance on distribution shifts compared to the fine-tuned model, but the WiSE-FT [25] curves corresponding to the averaged weights also demonstrate better ID/OOD trade-off than the WiSE-FT curve of the fine-tuned model, as illustrated in Fig. B. This indicates the importance of getting closer to the weight center, even for OOD datasets.

Another interesting point is that depending on the traits of datasets, the position of local minima differs. ImageNet-V2 has a similar dataset distribution to ImageNet since it shares the same data collection and categorization policy, and its local optima lies close to that of ImageNet. On the other hand, on the datasets with harsh variations (*e.g.*, ImageNet-A), the local minima are positioned much closer to the pre-trained model than the original ImageNet or ImageNet-V2. This loss landscape gives an intuitive insight into the similarity between OOD datasets and ImageNet.

In conclusion, there is no universal interpolation ratio optimal for every distribution shift. However, all the local minima lie on the line between the weight center and the pre-trained model. This implies the importance of proximity to the weight center in achieving a better WiSE-FT line.

D Detailed Proof of Model Stock

Here, we present detailed proof of Model Stock introduced in §3. We first show the case with two fine-tuned models and extend our proof toward N fune-tuned models.

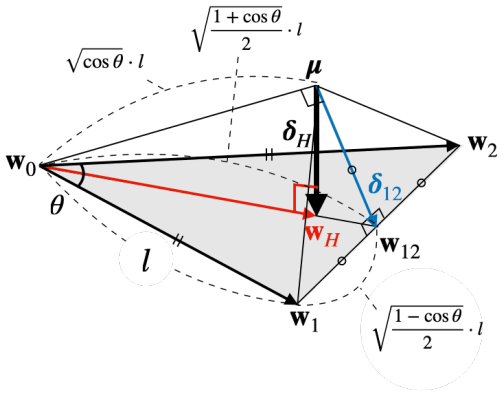
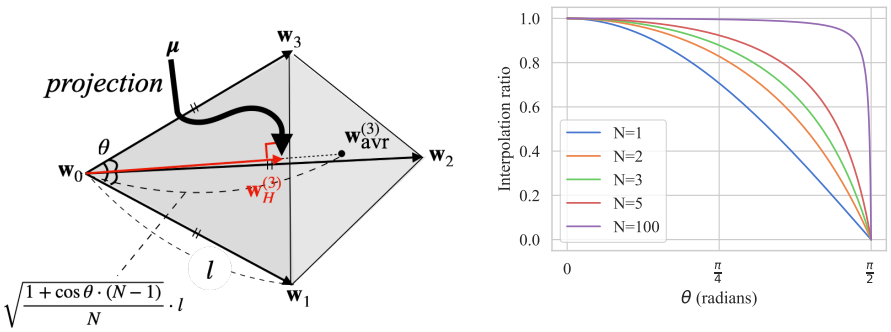


Fig. C: Model Stock with two fine-tuned models. We reference the illustration in Fig. 6 to more understandably substantiate merging two fine-tuned models.

On two fine-tuned models. We will prove step-by-step how the optimal interpolation ratio t in Eq. (2) in the main paper is derived. Using the same notation as in §3, we denote the magnitude and the angle between the fine-tuned weights as l and θ, respectively. Starting from the fact that $\triangle\boldsymbol{\mu}\mathbf{w}_1\mathbf{w}_2$ is a right isosceles triangle, we can derive the following relations from Fig. C:

$$\begin{aligned}
\overline{\mathbf{w}_{12}\mathbf{w}_1} &= \overline{\mathbf{w}_{12}\mathbf{w}_2} = \overline{\mathbf{w}_{12}\boldsymbol{\mu}} \\
&= \sqrt{\frac{1-\cos\theta}{2}}\cdot l \quad (\text{from } \triangle\mathbf{w}_0\mathbf{w}_1\mathbf{w}_2 \text{ and } \triangle\boldsymbol{\mu}\mathbf{w}_1\mathbf{w}_2) \qquad (1) \\
\Rightarrow \overline{\mathbf{w}_{12}\mathbf{w}_0} &= \sqrt{\overline{\mathbf{w}_1\mathbf{w}_0}^2 - \overline{\mathbf{w}_{12}\mathbf{w}_1}^2} \\
&= \sqrt{1^2 - \frac{1-\cos\theta}{2}}\cdot l \\
&= \sqrt{\frac{1+\cos\theta}{2}}\cdot l \quad (\text{from } \triangle\mathbf{w}_0\mathbf{w}_1\mathbf{w}_{12} \text{ and Eq. (1)}) \qquad (2) \\
\Rightarrow \overline{\mathbf{w}_0\boldsymbol{\mu}} &= \sqrt{\overline{\mathbf{w}_{12}\mathbf{w}_0}^2 - \overline{\mathbf{w}_{12}\boldsymbol{\mu}}^2} \\
&= \sqrt{\frac{1+\cos\theta}{2} - \frac{1-\cos\theta}{2}}\cdot l \\
&= \sqrt{\cos\theta}\cdot l \quad (\text{from } \triangle\mathbf{w}_0\boldsymbol{\mu}\mathbf{w}_{12}, \text{ Eq. (1) and Eq. (2)}) \qquad (3) \\
\Rightarrow t &:= \frac{\overline{\mathbf{w}_H\mathbf{w}_0}}{\overline{\mathbf{w}_{12}\mathbf{w}_0}} = \frac{\overline{\mathbf{w}_H\mathbf{w}_0}}{\overline{\mathbf{w}_0\boldsymbol{\mu}}}\cdot\frac{\overline{\mathbf{w}_0\boldsymbol{\mu}}}{\overline{\mathbf{w}_{12}\mathbf{w}_0}} \\
&= \left(\frac{\overline{\mathbf{w}_0\boldsymbol{\mu}}}{\overline{\mathbf{w}_{12}\mathbf{w}_0}}\right)^2 \quad (\text{from } \triangle\mathbf{w}_0\boldsymbol{\mu}\mathbf{w}_{12} \sim \triangle\mathbf{w}_0\mathbf{w}_H\boldsymbol{\mu}) \\
&= \frac{2\cos\theta}{1+\cos\theta} \quad (\text{from Eq. (2) and Eq. (3)}) \qquad (4)
\end{aligned}$$



(a) Model Stock with N fine-tuned models. **(b)** Variation of the interpolation ratio t.

Fig. D: Model Stock with N fine-tuned models and Interpolation Ratio Variation. (a) We visualize a special case of $N = 3$ (tetrahedron) for better understanding. (b) The trend towards $t = 1$ with increasing N illustrates that $\mathbf{w}_H^{(N)}$ on the N-dimensional simplex approaches $\mathbf{w}_{\text{avr}}^{(N)}$, reflecting a growing dependence on the number of fine-tuned models.

□

Interestingly, $\mathbf{w}_H$ is located at an orthocenter of the triangle $\triangle \mathbf{w}_0\mathbf{w}_1\mathbf{w}_2$ with the given optimal ratio t.

On N fine-tuned models. Similarly, we can derive a more generalized interpolation ratio for $N \geq 2$. Our goal is to find the weight $\mathbf{w}_{\text{avr}}^{(N)}$ that is on the hyper-plane spanned by $\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_N$ and closest to the weight center $\boldsymbol{\mu}$, as described in Fig. Da. Again, for simplicity, we treat $\mathbf{w}_0$ as the origin $\mathbf{O}$.

Based on the observation, we presume that the following two conditions hold:

$$\begin{cases} \mathbf{w}_H^{(N)} = t \cdot \mathbf{w}_{\text{avr}}^{(N)} \\ (\mathbf{w}_{\text{avr}}^{(N)} - \mathbf{w}_H^{(N)}) \cdot (\boldsymbol{\mu} - \mathbf{w}_H^{(N)}) = 0. \end{cases} \tag{5}$$

The first condition comes from the symmetry of an N-simplex structure, and the second condition holds since the orthogonal projection is the minimal distance from $\boldsymbol{\mu}$. Then, we can derive t as follows:

By substituting the first condition into the second condition from Eq. (5),

$$\begin{aligned} & (\mathbf{w}_{\text{avr}}^{(N)} - \mathbf{w}_H^{(N)}) \cdot (\boldsymbol{\mu} - \mathbf{w}_H^{(N)}) = 0 \\ \Rightarrow\ & \mathbf{w}_{\text{avr}}^{(N)} \cdot \boldsymbol{\mu} - t \cdot \|\mathbf{w}_{\text{avr}}^{(N)}\|^2 = 0 \\ \Rightarrow\ & t = \frac{\boldsymbol{\mu} \cdot \mathbf{w}_{\text{avr}}^{(N)}}{\|\mathbf{w}_{\text{avr}}^{(N)}\|^2}. \end{aligned} \tag{6}$$

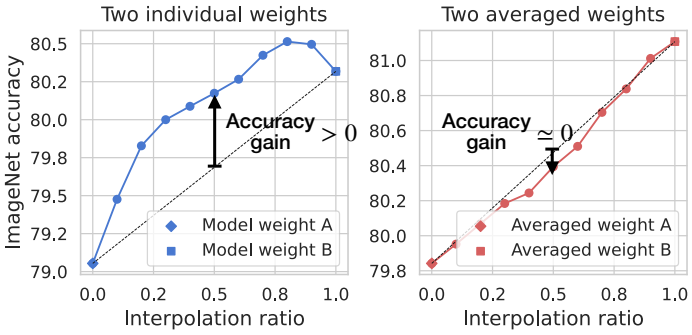


Fig. E: Ensembling impact disappears when interpolating between two averaged weights. We plot the ImageNet performance of interpolated weights between two selected fine-tuned models in Model Soup [24] (left) and between their corresponding weight centers (right).

Note that the norm of the N-averaged fine-tuned weights can be derived as follows:

$$\begin{aligned}\|\mathbf{w}_{\text{avr}}^{(N)}\|^2 &= \frac{1}{N^2}(\mathbf{w}_1 + \ldots + \mathbf{w}_N) \cdot (\mathbf{w}_1 + \ldots + \mathbf{w}_N) \\ &= \frac{1}{N^2}(l^2 + l^2 \cos\theta \cdot (N-1)) \cdot N \\ &= \frac{l^2}{N}(1 + \cos\theta \cdot (N-1)),\end{aligned} \tag{7}$$

while the term $\boldsymbol{\mu} \cdot \mathbf{w}_{\text{avr}}^{(N)}$ can be simplified as

$$\boldsymbol{\mu} \cdot \mathbf{w}_{\text{avr}}^{(N)} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{\mu} \cdot \mathbf{w}_i) = l^2 \cos\theta \quad \text{(from Lemma)}. \tag{8}$$

By substituting Eq. (7) and Eq. (8) into Eq. (6), we can finally derive the optimal interpolation ratio t as follows:

$$t = \frac{N\cos\theta}{1 + (N-1)\cos\theta} \quad \square \tag{9}$$

Fig. Db displays how the optimal interpolation ratio t varies as a function of θ with different numbers of fine-tuned models. As N increases, t trends towards 1, indicating that $\mathbf{w}_H^{(N)}$ on the N-dimensional simplex gets closer to $\mathbf{w}_{\text{avr}}^{(N)}$. This shows increasing dependence on fine-tuned models as their number grows.

E Discussion — Rethinking Pivotal Prior Studies

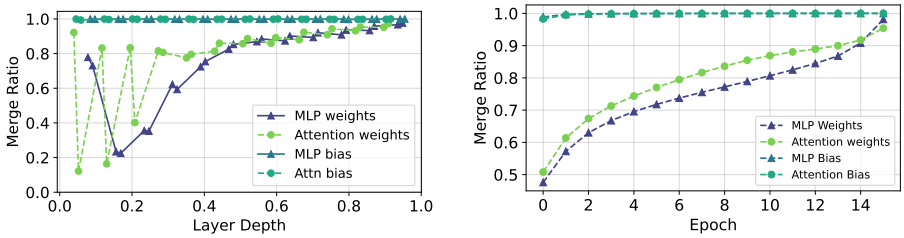
In this section, we extend our findings to reinterpret the underlying mechanics in prior studies, WiSE-FT [25] and Model Soups [24], through a consistent rationale to illuminate their effectiveness.

WiSE-FT [25] is a state-of-the-art robust fine-tuning method for CLIP-based models. It demonstrates that linearly combining weights of the pre-trained and fine-tuned models achieves significant accuracy gain on distribution shifts. We argue that the WiSE-FT model's superiority over a fine-tuned model can be interpreted by its weights being closer to the center of the corresponding weight distribution. Fig. 3 already showed fine-tuned models typically lie on the periphery of flat minima. Given that the angle $\angle \mathbf{w}_0 \mathbf{w}_{\text{avr}}^{(50)} \mathbf{w}_1$ is nearly a right angle, along the line $\overline{\mathbf{w}_0 \mathbf{w}_1}$, multiple weight points are closer to the center than a single fine-tuned model, thereby enhancing performance. Note that $\mathbf{w}_H$ is the closest to the center among the line $\overline{\mathbf{w}_0 \mathbf{w}_1}$. More discussions on performance boosts observed in distribution shifts are provided in the Appendix C.

Model Soup [24] merges various fine-tuned models' weights trained from varied hyper-parameters. It has been credited with delivering enhanced performance across ImageNet and distribution shifts. Here, we interpret the performance improvements of Model Soup as the result of the proximity to the center of weight distribution. Consider two weight vectors, $\mathbf{w}_A$ and $\mathbf{w}_B$, fine-tuned with different hyper-parameters and following Gaussian distribution $\mathcal{N}(\boldsymbol{\mu_A}, \Sigma_A)$ and $\mathcal{N}(\boldsymbol{\mu_B}, \Sigma_B)$ respectively. Then, the interpolated weight vector $\mathbf{w}_{AB} = t \cdot \mathbf{w}_A + (1-t) \cdot \mathbf{w}_B$ also follows a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu_{AB}}, \Sigma_{AB})$. The expected squared distance from the interpolated weight vector to its mean $\boldsymbol{\mu_{AB}}$ is minimized to $\frac{\text{trace}(\Sigma_A)\text{trace}(\Sigma_B)}{\text{trace}(\Sigma_A)+\text{trace}(\Sigma_B)}$ when t is chosen to $\frac{\text{trace}(\Sigma_B)}{\text{trace}(\Sigma_A)+\text{trace}(\Sigma_B)}$, indicating the reduction of variance through weight interpolation (*i.e.*, the distance between $\mathbf{w}_{AB}$ and $\boldsymbol{\mu_{AB}}$ might be closer than each weight's distance). For example, if $\text{trace}(\Sigma_B)$ is equal to $\text{trace}(\Sigma_A)$, this minimum squared distance is exactly half of the sum of the individual traces when $t = 0.5$. This insight suggests that the performance gains realized by Model Soup could be due to reduced variance resulting from merging numerous weights.

We set up a toy experiment to evaluate the effect of variance reduction in the Model Soup scenario by comparing the interpolation of fine-tuned weights with the interpolation of their corresponding weight centers, when $N = 2$. In the former case, variance reduction exists along with the effect of merging diverse hyper-parameters, while in the latter case, performance gain would only come from hyper-parameter diversity. If the diversity of hyper-parameters is a major factor, the performance gain from interpolation of central weights should remain the same. To test this, we assessed the ImageNet performance of interpolated weights between pairs of fine-tuned models within Greedy Model Soup[1] [24] and compared it to interpolations between their central weights, calculated as the average of 20 differently seeded models. Fig. E shows that, unlike interpolations between individual models, using the centers does not significantly improve performance. This suggests that proximity to the center of the weight distribution may play a more critical role than hyper-parameter diversity in weight ensemble methods in this case.

[1] We opt for Greedy Model Soup to show that even the interpolation of models from the best merging combination does not benefit from the impact of weight diversity.



(a) Layer depth vs. interpolation ratio t. **(b) Epoch vs. interpolation ratio t.**

Fig. F: Trend of interpolation ratio t during Model Stock training.

It is also worth noting that $\boldsymbol{\mu_{AB}}$ always surpasses the performance of $\mathbf{w_{AB}}$ for the same interpolation ratio t, indicating that the importance of proximity to the center remains consistent for interpolated weights. With extensive future research, this understanding could provide valuable insights for developing more generalizable and effective weight-merging techniques.

F Analysis of the interpolation ratio t

We analyze the interpolation ratio $t = \frac{2\cos\theta}{1+\cos\theta}$ in a layer-wise manner. During a Model Stock experiment on CLIP ViT-B/32 with 16 epoch training, we log the layer-wise merge ratios at every merging period. Figure F visualizes the averaged interpolation ratio during Model Stock training. We plot two trends of the interpolation ratio for the layer depth and training step. Our overall observation indicates the bias layers have high merge ratios t ($\simeq 1$) with small angles θ ($\simeq 0$), implying that the bias layers do not need to enjoy the pre-trained model, similar to our discussion in §2 and §3. Focusing on the weight layers, Figure Fa shows a U-shape tendency as the layer depth increases, implying the weights of intermediate layers can be more diverse (*i.e.*, larger angle θ) than those of early and later layers. Our intuition here is that since the early and later layers are directly connected to input data and output labels, respectively, they may not demand the advantage of the pre-trained weight. Figure Fb presents that the models at the early training stage are more diverse and they enjoy the pre-trained weights more than those of the later training stage. As the model approaches convergence, the diversity of fine-tuning models decreases (*i.e.*, smaller angle θ).

G Experimental setup

Here, we present detailed setups for the experiments in §4. We utilize AdamW optimizer [15] with a weight decay of 0.1. We employ two training setups for Model Stock. The first is training Model Stock with a learning rate of 3×10^{-5} in 10 epochs with minimal data augmentation. The minimal data augmentation utilizes random resize crop augmentation with a minimum crop ratio of 0.9,

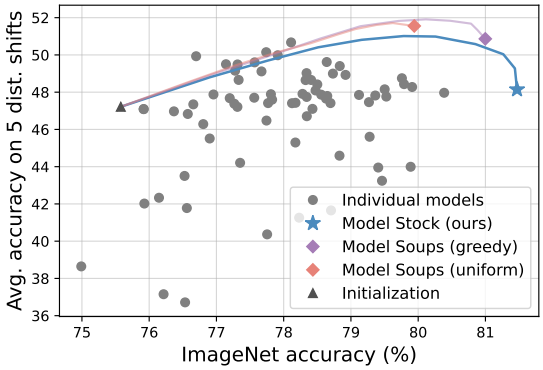


Fig. G: Results on LP initialization. We plot in-distribution ImageNet accuracy (x-axis) and distribution shift results (y-axis) with individual fine-tuned models (gray circles) and Model Soups [24]. Note that Model Stock has much smaller (35× smaller) computational costs than Model Soups, leveraging 71 various fine-tuned models as in the original paper.

mixup [27] augmentation with β=0.5, following Model Soup's "standard grid search" setting. The other is training Model Stock with a learning rate of 2×10^{-5} in 16 epochs with strong data augmentation. The strong data augmentation utilizes random resize crop augmentation with a minimum crop ratio of 0.08 and random augmentation [3] ($N = 2$, $M = 10$) following Model Soup's "random search" setting. When experimenting with the ViT-B/16 and ViT-L/14 models, we adjusted the learning rate and batch size to accommodate the GPU memory constraints.

H Additional Experiments

We present additional experimental studies to verify the effectiveness and applicability of Model Stock.

H.1 Experiments with LP initialization

We conduct Model Stock with LP initialization and compare it with Model Soups that are initialized from LP. The results are in Fig. G. In this experiment, we use the 16-epoch training setup with strong data augmentation for training Model Stock. As shown in Fig. G, Model Stock outperforms the individual fine-tuned models[2] (gray dots) on ImageNet accuracy. Model Stock also demonstrates competitive performance against Model Soups considering WiSE-FT curves. Note that Model Stock is much more efficient (35×) than Model Soups, which utilize 71 models in this experiment.

[2] All the individual model checkpoints are from the official Model Soup repository.

Table A: Complete results of Table 3 with ObjectNet [1] and ImageNet-ReaL [2].

Method	In-distribution		Distribution shifts				
	ImageNet	IN-ReaL	IN-V2	IN-R	IN-A	IN-Sketch	ObjectNet
Zero-shot	68.3	75.1	62.0	**77.7**	49.9	48.3	54.2
Vanilla FT	82.8	87.8	72.9	66.4	43.7	48.0	51.8
Vanilla FT*	83.7	87.8	73.5	67.6	40.0	48.6	50.1
LP [10]	79.7	-	71.5	52.4	27.8	40.5	-
LP-FT [10]	81.7	-	71.6	72.9	49.1	48.4	-
CAR-FT [16]	83.2	-	73.0	71.3	43.7	49.5	-
FTP [22]	84.2	-	74.6	47.2	26.5	50.2	-
FLYP [5]	82.6	-	73.0	71.4	48.1	49.6	**58.7**
Lipsum-FT [17]	83.3	-	73.6	75.9	49.9	51.4	54.4
CaRot [18]	83.1	-	74.1	**77.7**	**51.6**	**52.7**	56.6
Model Stock	84.1	88.8	74.8	71.8	51.2	51.8	55.0
Model Stock*	**85.2**	**89.1**	**75.3**	68.7	45.0	51.3	52.3

Table B: Comparison against Model Soups [24] on CLIP ViT-B/16. Model Stock shows comparable performance with Model Soups.

Method	ImageNet	Avg. shifts
CLIP zero-shot Init.	68.3	58.4
Vanilla FT	82.8	56.6
Vanilla FT*	83.7	55.9
Uniform Model Soup	84.4	**62.7**
Greedy Model Soup	84.3	60.4
Model Stock	84.1	61.0
Model Stock*	**85.2**	58.5

H.2 Complete comparison results on CLIP ViT-B/16

In the main paper, we omit the results of ObjectNet [1] on CLIP ViT-B/16 experiments since the comparison methods such as LP-FT [10], FTP [22] have not evaluated on ObjectNet benchmark. We here show the results with ObjectNet [1] and ImageNet-ReaL [2] of CLIP ViT-B/16 in Table A. We additionally compare Model Stock with recent fine-tuning methods including FLYP [5], Lipsum-FT [17], and CaRot [18] Model Stock consistently demonstrates its effectiveness with ObjectNet and ImageNet-ReaL as well.

H.3 Model Stock vs. Model Soups on CLIP ViT-B/16

Table B shows the performance of Model Stock on the pretrained CLIP ViT-B/16 model. Since the original Model Soups paper [24] only provides CLIP ViT-B/32 models, we replicate Model Soups experiments on CLIP ViT-B/16. We fine-tuned 48 models from CLIP ViT-B/16 initialization following the standard grid hyper-parameter sweep (*i.e.*, *zero-shot* initialization setting). Model Stock shows

Table C: Model Stock with different hyper-parameters on CLIP ViT-B/32.

Method	ImageNet	Avg. shifts
Model Stock	79.89	50.99
Model Stock w/ different hyper-parameters	79.75±0.45	50.40±0.84

Table D: Performance comparison of merging units in Model Stock. This table presents the overall performance of Model Stock using different merging units: entire weight merging, entire weight merging based on transformer block angle, layer-wise merging, and filter-wise merging. It highlights the effectiveness of each strategy in approaching the weight center and their impact on the model's performance.

Merging Unit	Target		Avg. Shifts
	IN	IN-ReaL	
Entire weights	79.69	85.39	46.40
Entire weights (rep. blocks only)	79.64	85.38	48.28
Layer-wise (ours)	**80.12**	85.65	**48.84**
Filter-wise	80.10	**85.67**	48.72

comparable performance against Model soups. Note that Model Soups requires 24× more training cost than Model Stock.

H.4 Model Stock with different hyper-parameters

To verify the validity of Model Stock beyond the setup of the main paper (*i.e.*, different random seeds with the same hyper-parameters), we conduct Model Stock with different hyper-parameters. In detail, when we fine-tune two models for Model Stock, we choose different hyper-parameter for each model (*e.g.*, learning rate, data augmentation.). To ensure the basic assumption of Model Stock, we use the same batch size and training epochs. C shows the experimental results on CLIP ViT-B/32. We repeat 5 runs and report accuracy with standard deviation. Model Stock with different hyper-parameters shows comparable performance to the original one.

H.5 Ablation study on merging unit

We investigate the efficacy of different merging units within our method, Model Stock. Our default approach employs layer-wise merging, but alternatives include merging based on the angle between 1) entire weights, 2) weights of the entire repetitive transformer blocks following [24], or 3) using a filter-wise approach as discussed in §A.3. The results of these ablations are summarized in Table D, where we assess the overall performance based on the chosen merging unit.

Our analysis reveals that the accuracy of noise distribution estimation is critical in approaching the weight center. When assuming weight noise across the

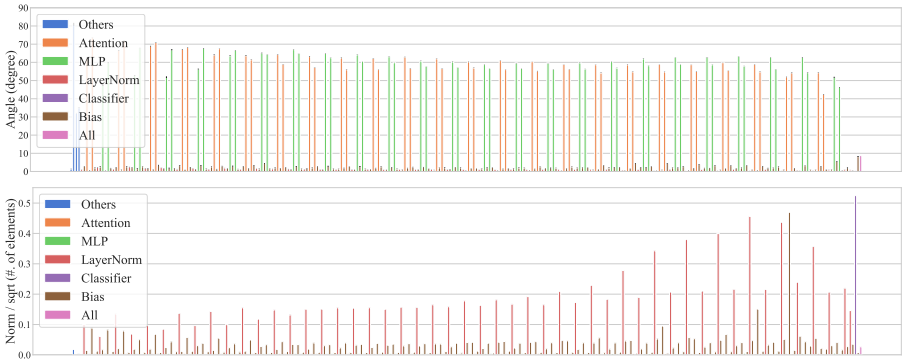
entire model, our method does not approximate the weight center as effectively as it does with layer-wise merging, leading to suboptimal overall performance. Similarly, the merging performance based on the angle of transformer blocks was insufficient. Conversely, while filter-wise noise demonstrates a larger standard deviation in angle, as depicted in Fig. N, this increased variance results in a more significant error in Gaussian distribution approximation. Consequently, the overall performance under filter-wise merging is slightly inferior to layer-wise one.

These findings underscore the importance of accurately modeling noise distribution in enhancing the performance of Model Stock. As our understanding and ability to model this noise distribution improve, we anticipate further increases in the efficacy and robustness of our approach.

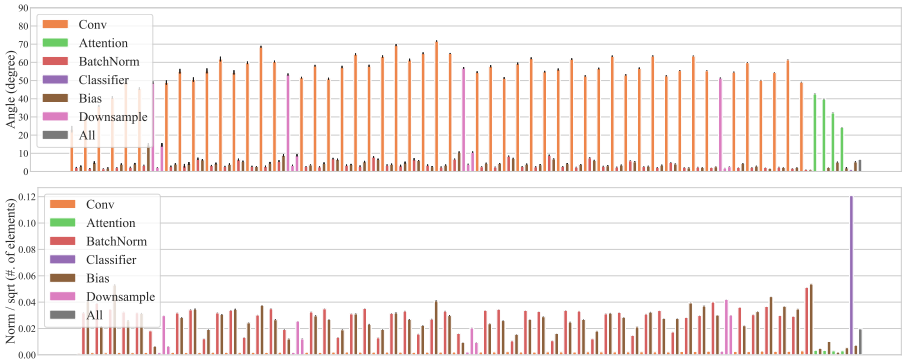
References

1. Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., Katz, B.: Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. NeurIPS (2019)
2. Beyer, L., Hénaff, O.J., Kolesnikov, A., Zhai, X., Oord, A.v.d.: Are we done with imagenet? arXiv preprint arXiv:2006.07159 (2020)
3. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: CVPRW. pp. 702–703 (2020)
4. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
5. Goyal, S., Kumar, A., Garg, S., Kolter, Z., Raghunathan, A.: Finetune like you pretrain: Improved finetuning of zero-shot vision models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19338–19347 (2023)
6. He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., Neubig, G.: Towards a unified view of parameter-efficient transfer learning. arXiv preprint arXiv:2110.04366 (2021)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
8. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
9. Krizhevsky, A.: Learning multiple layers of features from tiny images. In: Tech Report (2009)
10. Kumar, A., Raghunathan, A., Jones, R., Ma, T., Liang, P.: Fine-tuning can distort pretrained features and underperform out-of-distribution. arXiv preprint arXiv:2202.10054 (2022)
11. Lee, Y., Chen, A.S., Tajwar, F., Kumar, A., Yao, H., Liang, P., Finn, C.: Surgical fine-tuning improves adaptation to distribution shifts. In: ICLR (2022)
12. Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T.: Visualizing the loss landscape of neural nets. NeurIPS **31** (2018)
13. Lian, D., Zhou, D., Feng, J., Wang, X.: Scaling & shifting your features: A new baseline for efficient model tuning. NeurIPS **35**, 109–123 (2022)

14. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: CVPR. pp. 11976–11986 (2022)
15. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
16. Mao, X., Chen, Y., Jia, X., Zhang, R., Xue, H., Li, Z.: Context-aware robust fine-tuning. IJCV (12 2023). https://doi.org/10.1007/s11263-023-01951-2
17. Nam, G., Heo, B., Lee, J.: Lipsum-ft: Robust fine-tuning of zero-shot models using random text guidance. In: The Twelfth International Conference on Learning Representations (2024)
18. Oh, C., Lim, H., Kim, M., Han, D., Yun, S., Choo, J., Hauptmann, A., Cheng, Z.Q., Song, K.: Towards calibrated robust fine-tuning of vision-language models (2024), https://arxiv.org/abs/2311.01723
19. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
20. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. pp. 1–9 (2015)
21. Tian, J., He, Z., Dai, X., Ma, C.Y., Liu, Y.C., Kira, Z.: Trainable projected gradient method for robust fine-tuning. In: CVPR. pp. 7836–7845 (2023)
22. Tian, J., Liu, Y.C., Smith, J.S., Kira, Z.: Fast trainable projection for robust fine-tuning. In: NeurIPS (2023)
23. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jegou, H.: Training data-efficient image transformers &distillation through attention. In: ICML. vol. 139, pp. 10347–10357 (July 2021)
24. Wortsman, M., Ilharco, G., Gadre, S.Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A.S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al.: Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In: ICML. pp. 23965–23998. PMLR (2022)
25. Wortsman, M., Ilharco, G., Kim, J.W., Li, M., Kornblith, S., Roelofs, R., Lopes, R.G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al.: Robust fine-tuning of zero-shot models. In: CVPR. pp. 7959–7971 (2022)
26. Zaken, E.B., Ravfogel, S., Goldberg, Y.: Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. arXiv preprint arXiv:2106.10199 (2021)
27. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. ICLR (2018)



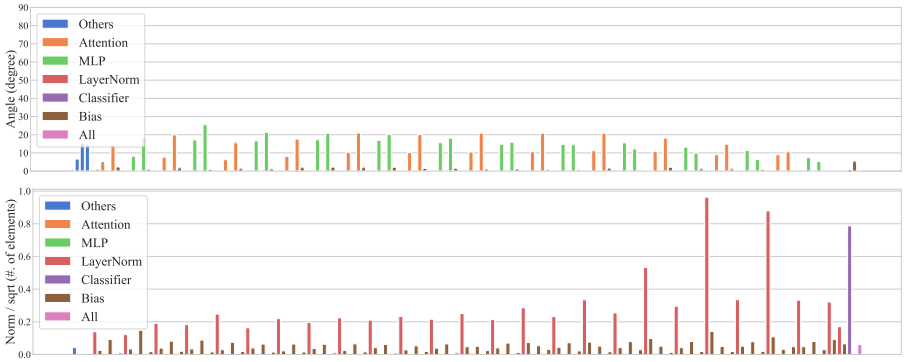
(a) CLIP ViT-L/14



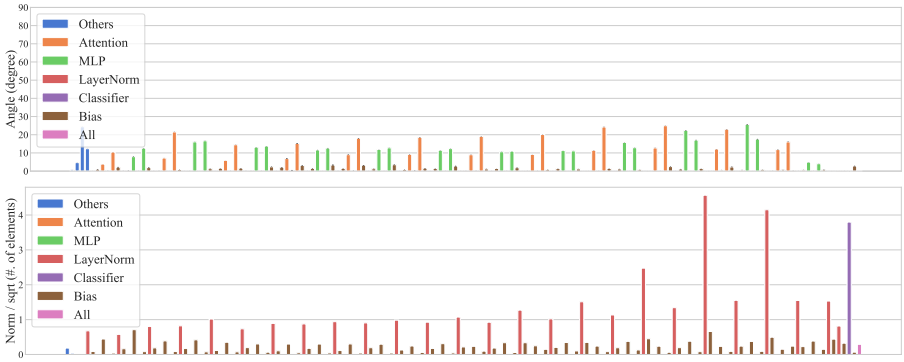
(b) CLIP ResNet50

(c) OpenCLIP ConvNeXt

Fig. H: Layer-wise angle and norm across different model architectures. The angle and norm for CLIP ViT-L/14, CLIP ResNet50, and OpenCLIP ConvNeXt are displayed from top to bottom. These metrics demonstrate consistency regardless of the model type from left (first layer) to right (last layer). It is important to note that we also depict the error bars for each layer in all figures, but they are not visible in most layers due to the small standard deviation.



(a) SGD optimizer

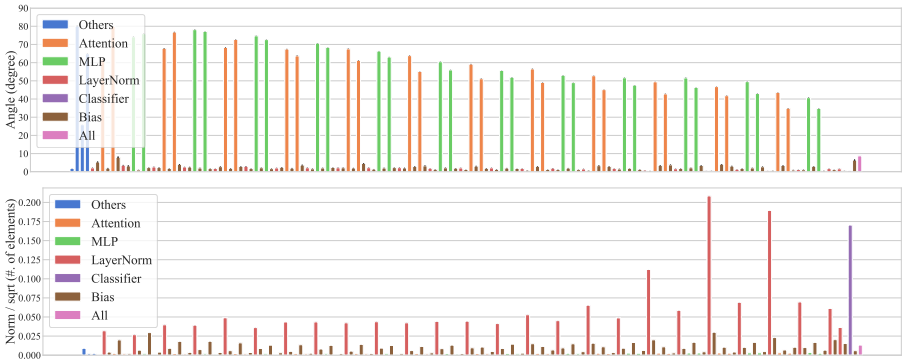


(b) SGD optimizer with momentum

Fig. I: Layer-wise angle and norm across different optimizers. Displayed from top to bottom are the angle and norm for models trained with SGD and SGD with momentum, respectively. These metrics demonstrate consistency regardless of the optimization strategy from left (first layer) to right (last layer).

(a) Vanilla model (10 epochs + no augmentation)

(b) + longer epochs (16 epochs)



(c) + RRC

Fig. J: Layer-wise angle and norm across different augmentations. Displayed from top to bottom are the angle and norm for the vanilla model (10 epochs + no augmentation), +longer epochs (16 epochs), and +RRC. Each augmentation is applied incrementally. These metrics demonstrate consistency regardless of the augmentations from left (first layer) to right (last layer).

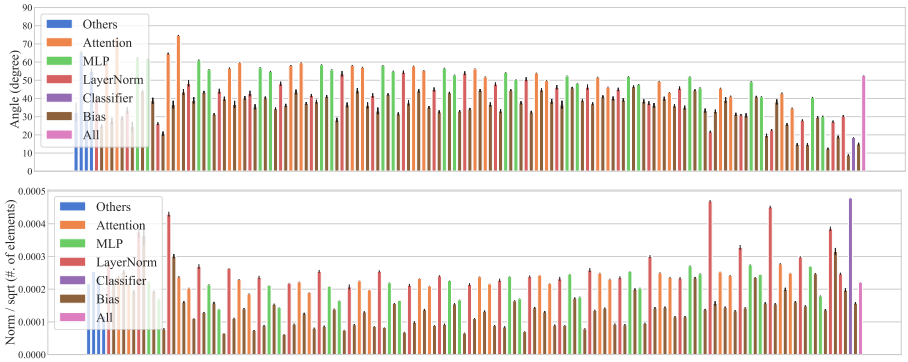


Fig. K: Layer-wise angle and norm across different datasets. The angle and norm for models trained on different datasets, including CIFAR [9] are displayed from top to bottom. These metrics demonstrate consistency regardless of the dataset type from left (first layer) to right (last layer).

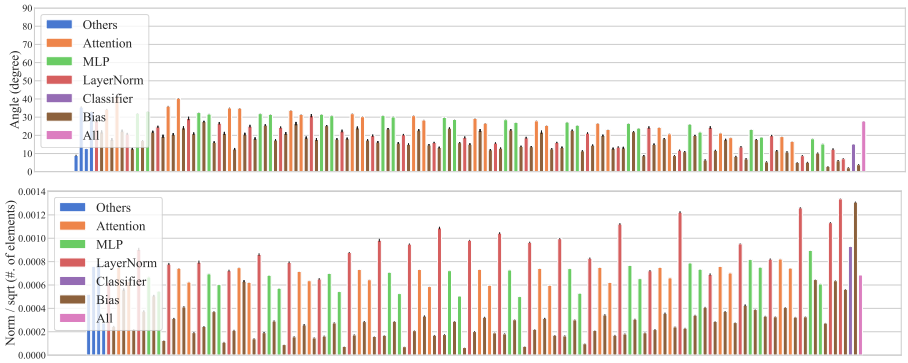


Fig. L: Layer-wise angle and norm across different classifier initializations. The angle and norm for models trained with differently initialized networks following the LP-FT [10] method are displayed from top to bottom. These metrics demonstrate consistency regardless of the initialization method from left (first layer) to right (last layer).

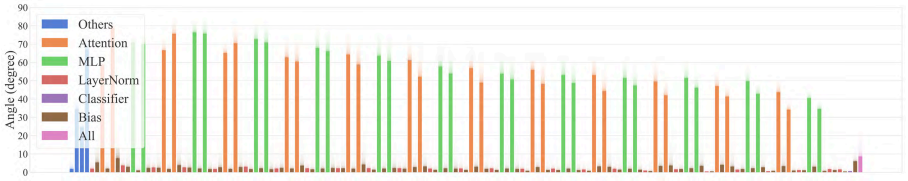
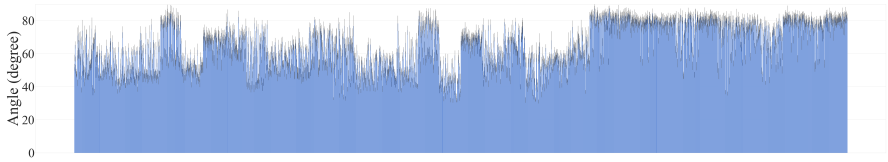
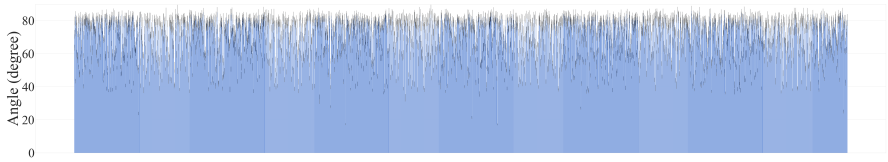


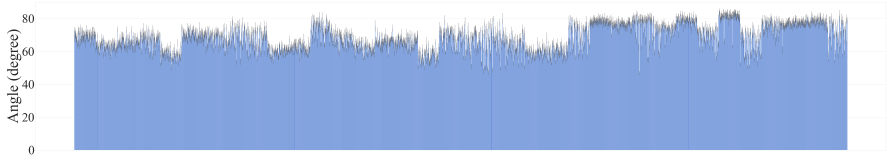
Fig. M: Layer-wise angle during training. Displayed are the overlapped angles across models trained with different random seeds at each timestamp. Even during training, the angle remains highly consistent, decreasing as training progresses.



(a) Filter-wise angle between attention weights in the first transformer block of ViT-B/32



(b) Filter-wise angle between MLP weights in the first transformer block of ViT-B/32



(c) Filter-wise angle between attention weights in the second transformer block of ViT-B/32

Fig. N: Filter-wise angle for attention and MLP layers in ViT-B/32. We display filter-wise angles for each layer. Each bar represents each row (*i.e.*, filter) in the given layer. Interestingly, the angles between the filters of the fine-tuned weights exhibit similar values, while the standard deviation between each filter is notably larger than that of the angle between each layer. Due to the large number of layers, only representative layers are selected for display.

Fig. O: Layer-wise angle and norm for DeiT. The angle and norm for DeiT-base models are displayed, each trained with different random seeds. These models are initially pre-trained on ImageNet-21K [19] and then fine-tuned on ImageNet-1K. The consistency observed in the metrics is maintained even in the DeiT training setting.