Robust fine-tuning of zero-shot models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large pre-trained models such as CLIP offer consistent accuracy across a range of data distributions when performing zero-shot inference (i.e., without fine-tuning on a specific dataset). Although existing fine-tuning approaches substantially improve accuracy in-distribution, they also reduce out-of-distribution robustness. We address this tension by introducing a simple and effective method for improving robustness: ensembling the weights of the zero-shot and fine-tuned models (WiSE-FT). Compared to standard fine-tuning, WiSE-FT provides large accuracy improvements out-of-distribution, while matching or improving in-distribution accuracy. On ImageNet (in-distribution) and five derived distribution shifts, WiSE-FT improves out-of-distribution accuracy by 2 to 10 percentage points (pp) while increasing in-distribution accuracy by nearly 1 pp relative to standard fine-tuning. WiSE-FT achieves similarly large robustness improvements (2 to 15 pp) on a diverse set of six further distribution shifts, and in-distribution accuracy gains of 0.8 to 3.3 pp compared to standard fine-tuning on seven commonly used transfer learning datasets. These improvements come at no additional computational cost during fine-tuning or inference.

1 Introduction

A foundational goal of machine learning is to develop models that work reliably across a broad range of data distributions. Recently, large pre-trained models such as CLIP [72] and ALIGN [41] have demonstrated unprecedented robustness to challenging distribution shifts where prior robustness interventions failed to improve performance [85]. While these results point towards pre-training on large, heterogeneous datasets as a promising direction for increasing robustness, an important caveat is that these robustness improvements occur only in the zero-shot setting, i.e., when the model performs inference without fine-tuning on a specific target distribution [72].

In a concrete application, a zero-shot model can be fine-tuned on extra application-specific data, which often yields large performance gains on the target distribution. However, Radford et al. [72] have shown that current fine-tuning techniques carry a hidden cost: the out-of-distribution accuracy of their fine-tuned CLIP models is often lower than that of the original zero-shot model. This leads to a natural question: *Can zero-shot models be fine-tuned without reducing out-of-distribution accuracy?*

As pre-trained models are becoming a cornerstone of machine learning, techniques for fine-tuning them on downstream applications are increasingly important. Indeed, the question of robustly fine-tuning pre-trained models has recently also been raised as an open problem by several authors [3, 9, 72]. Andreassen et al. [3] explored several fine-tuning approaches but found that none yielded models with improved robustness at high accuracy. Furthermore, Taori et al. [85] demonstrated that no current algorithmic robustness interventions provide consistent gains across the distribution shifts where zero-shot CLIP excels.

In this paper, we introduce a robust way of fine-tuning zero-shot models that achieves the best of both worlds: increased performance out-of-distribution while maintaining or even improving in-distribution accuracy relative to standard fine-tuning. Our method (Figure 1) has two steps: first,

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

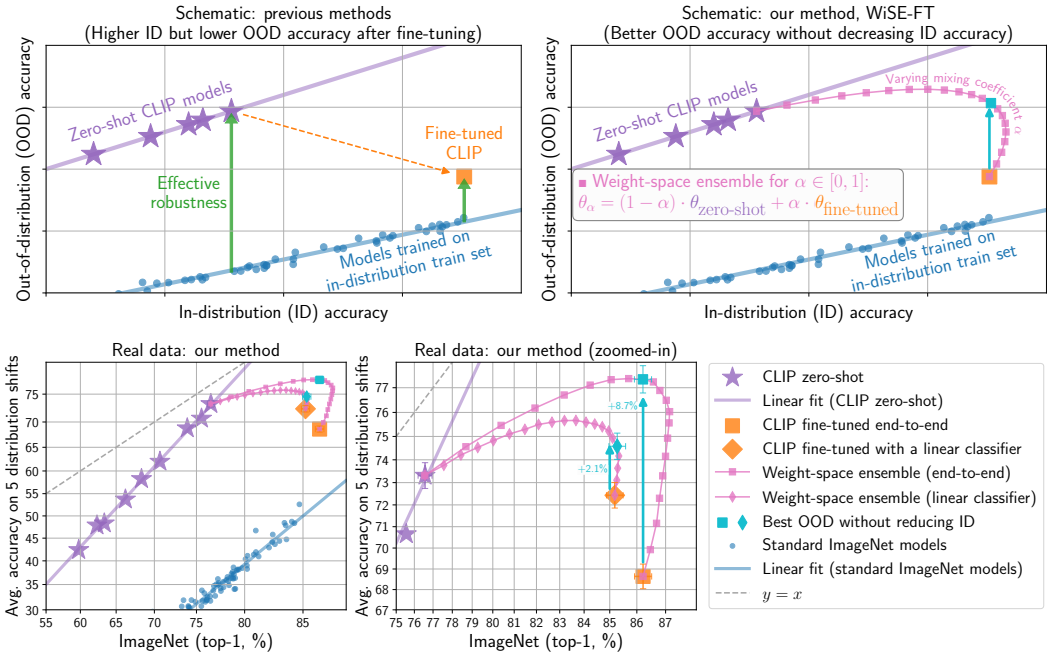

Figure 1: Compared to standard fine-tuning, weight-space ensembles for fine-tuning (WiSE-FT) improve out-of-distribution (OOD) accuracy without decreasing in-distribution (ID) performance. **(Top left)** Zero-shot CLIP modelsexhibit high effective robustness and moderate in-distribution accuracy, while standard fine-tuning—either end-to-end or with a linear classifier (final layer)—attains higher ID accuracy and less effective robustness. **(Top right)** Our method linearly interpolates between the zero-shot and fine-tuned models with a mixing coefficient $\alpha \in [0, 1]$. **(Bottom)** On five distribution shifts derived from ImageNet (ImageNetV2, ImageNet-R, ImageNet Sketch, ObjectNet, and ImageNet-A), WiSE-FT improves average OOD accuracy by 8.7 percentage points (pp) when fine-tuning end-to-end (+2.1 pp when fine-tuning a linear classifier) while maintaining ID accuracy.

we fine-tune the zero-shot model on application-specific data. Second, we combine the original zero-shot and fine-tuned models by linearly interpolating between their weights, which we refer to as weight-space ensembling.

Compared to standard fine-tuning, weight-space ensembles for fine-tuning (WiSE-FT) substantially improve out-of-distribution accuracy without decreasing in-distribution performance. Concretely, on ImageNet [17] and five of the natural distribution shifts studied by Radford et al. [72], WiSE-FT applied to standard end-to-end fine-tuning improves out-of-distribution accuracy by 2 to 15 percentage points (pp) while maintaining the in-distribution accuracy of the fine-tuned model. Relative to the zero-shot model, WiSE-FT improves out-of-distribution accuracy by 1 to 9 pp. Moreover, WiSE-FT improves over a range of alternative approaches such as regularization and evaluating at various points throughout fine-tuning. The robustness comes at no additional computational cost during fine-tuning or inference because the zero-shot and fine-tuned models are ensembled into a single model of the same size.

To understand the robustness gains of WiSE-FT, we empirically analyze ensembling through the lens of distributional robustness. First, we study WiSE-FT when fine-tuning a linear classifier (last layer) as it is amenable to analysis: our procedure is then equivalent to ensembling the outputs of two models, and experiments point towards the complementarity of model predictions as a key property. We illustrate via detailed measurements how the predictions of zero-shot and fine-tuned models are diverse, and that models are more confident on the parts of the test distributions they perform well on.

For end-to-end fine-tuning, we connect our observations to earlier work on the phenomenology of deep learning. Neyshabur et al. [65] found that end-to-end fine-tuning the same model twice yielded two different solutions that were connected via a linear path in weight space along which error remains low, known as linear mode connectivity [25]. The authors therefore concluded that the two solutions are in the same basin of the loss landscape. Interestingly, linear interpolation in weight space succeeds despite non-linearity in the activation functions of the neural networks. Our

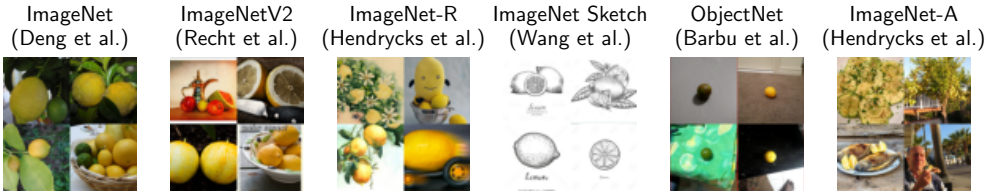

Figure 2: Samples of the class *lemon*, from ImageNet (in-distribution) and the derived out-of-distribution datasets considered in our main experiments.

observations suggest a similar phenomenon along the path generated by WiSE-FT, but the exact shape of the loss landscape and connection between in- and out-of-distribution error are still an open problem.

In addition to the aforementioned ImageNet distribution shifts, WiSE-FT consistently improves robustness on a diverse set of six further distribution shifts including: (i) geographic shifts in satellite imagery and wildlife recognition (WILDS-FMoW, WILDS-iWildCam) [44, 13, 6], (ii) reproductions of the popular image classification dataset CIFAR-10 with a distribution shift (CIFAR-10.1 and CIFAR-10.2) [73, 55], and (iii) datasets with distribution shift induced by temporal perturbations in videos (ImageNet-Vid-Robust and YTBB-Robust) [78]. The performance improvements are of similar magnitude, ranging from 2 to 15 pp.

Beyond the robustness perspective, WiSE-FT also improves in-distribution performance compared to standard fine-tuning, reducing the relative error rate by 6 to 62% on a range of seven datasets: ImageNet, CIFAR-10, CIFAR-100 [49], Describable Textures [14], Food-101 [10], SUN397 [91], and Stanford Cars [48]. Even when fine-tuning data is scarce—reflecting many application scenario—we find that WiSE-FT substantially improves in-distribution performance. When five examples per class are used for fine-tuning, WiSE-FT improves in-distribution performance by 0.3 to 6.1 percentage points compared to the best of the zero-shot and fine-tuned models on the seven aforementioned datasets. Overall, we find that WiSE-FT is universally applicable in the problems we studied, and we encourage its adoption for fine-tuning zero-shot models.

2 Background and experimental setup

Our experiments compare the performance of a zero-shot model, the corresponding fine-tuned model and models produced by WiSE-FT. Primarily, we contrast model accuracy on data from two related but different distributions $\mathcal{D}$ and $\mathcal{D}'$, where the expectation is that a robust model achieves consistent performance on both. We assume both distributions have test sets for evaluation, and $\mathcal{D}$ has an associated training set $\mathcal{S}_{\mathcal{D}}^{\text{tr}}$ which is typically used for training or fine-tuning. We borrow conventional nomenclature in referring to $\mathcal{D}$ as in-distribution (ID) and $\mathcal{D}'$ as out-of-distribution (OOD), even when evaluating models which have never been trained on ID data (e.g., CLIP zero-shot). Our goal is both high accuracy and consistent performance on these two distributions. This is a natural target as humans often achieve similar accuracy across distribution shifts [79].

For a model f we let $\mathsf{Acc}_{\mathcal{D}}(f)$ and $\mathsf{Acc}_{\mathcal{D}'}(f)$ refer to classification accuracy on the ID and OOD test sets respectively. We consider k-way image classification where x_i is an image with corresponding label $y_i \in \{1, ..., k\}$. The outputs of f are k-dimensional vectors of non-normalized class scores.

Distribution shifts. Distribution shifts can be broadly characterized into a) synthetic, e.g. ℓ_∞-adversarial examples or artificial changes in image contrast, brightness, etc. [33, 8, 7, 29, 2]; and b) natural, where samples are not perturbed after acquisition and changes in data distributions arise through naturally occurring variations in lighting, geographic location, crowdsourcing process, image styles, etc. [85, 73, 35, 36, 44]. Our focus is natural distribution shifts, which are more likely to occur in the real world. Specifically, our key results are shown for five distribution shifts illustrated in Figure 2 where $\mathcal{S}_{\mathcal{D}}^{\text{tr}}$ is ImageNet. More information on the ImageNet distribution shifts can be found in Appendix C.4 while related work can be found in Appendix E.

Effective robustness and scatter plots. Effective robustness scatter plots are central to our analysis, which illustrate model performance under distribution shift [73, 85]. These scatter plots display ID accuracy on the x-axis and OOD accuracy on the y-axis—a model f is shown as a point $(\mathsf{Acc}_{\mathcal{D}}(f),\ \mathsf{Acc}_{\mathcal{D}'}(f))$. Figure 1 exemplifies these scatter plots with both schematics and real

	IN (ID)	OOD datasets IN-V2	IN-R	IN-Sketch	ObjectNet*	IN-A	Avg OOD	Avg ID,OOD
CLIP zero-shot [72]	76.2	70.1	88.9	60.2	70.0	77.2	73.3	74.8
CLIP fine-tuned LC [72]	85.4	75.9	84.2	57.4	66.2	75.3	71.8	78.6
CLIP zero-shot (PyTorch)	76.6	70.5	89.0	60.9	69.1	77.7	73.4	75.0
CLIP fine-tuned LC (ours)	85.2	75.8	85.3	58.7	67.2	76.1	72.6	78.9
CLIP fine-tuned E2E (ours)	86.2	76.8	79.8	57.9	63.3	65.4	68.6	77.4
WiSE-FT (ours)								
LC, α=0.75	85.1	76.8	88.4	61.9	69.7	78.9	75.1	80.1
LC, α=0.4	82.7	75.8	89.7	63.0	70.7	79.6	75.8	79.2
LC, optimal α	85.3	76.9	89.8	63.0	70.7	79.7	76.0	80.7
E2E, α=0.75	87.0	78.8	86.1	62.5	68.1	75.2	74.1	80.5
E2E, α=0.4	86.2	79.2	89.9	**65.0**	71.9	80.7	77.3	81.8
E2E, optimal α	87.1	79.5	90.3	**65.0**	**72.1**	81.0	**77.6**	**82.3**

Table 1: ID and OOD performance of various methods. E2E: end-to-end; LC: linear classifier. The highest overall accuracy for each dataset is in bold, while the highest accuracy among models derived from CLIP is underlined. *Avg OOD* displays the mean performance among the five OOD datasets, while *Avg ID,OOD* shows the average of ImageNet (ID) and Avg OOD.

data. For more information including a complete definition of effective robustness [85] refer to Appendix A.2.

WiSE-FT Our proposed method, WiSE-FT, consists of two steps. First, we fine-tune the zero-shot model on application-specific data. Second, we combine the original zero-shot and fine-tuned models by linearly interpolating between their weights, also referred to as weight-space ensembling. Concretely, if θ_0 are the parameters of the zero-shot model and θ_1 are the parameters of the fine-tuned model, then WiSE-FT makes predictions via $f(x, (1-\alpha) \cdot \theta_0 + \alpha \cdot \theta_1)$ for a mixing coefficient $\alpha \in [0, 1]$. Refer to Appendix B for motivation and detail.

3 Results

Main results: ImageNet and associated distribution shifts. Tables 1 presents our main results on ImageNet and five derived distribution shifts. As illustrated in Figure 1, when the mixing coefficient α varies from 0 to 1, WiSE-FT simultaneously improves both ID and OOD accuracy (a breakdown for each dataset is shown in Appendix C.1). Table 1 shows that WiSE-FT improves OOD performance by 2 to 15 percentage points (pp) compared to standard fine-tuning, without reducing ID performance (α=0.4, end-to-end). Fixing α=0.4 is close to optimal for all OOD datasets, with only a 0.3 pp difference on average. Alternatively, when α=0.75, ImageNet performance improves by 0.9 pp while OOD performance increases by 2 to 10 pp compared to standard end-to-end fine-tuning. Appendix Table 2 offers an alternative perspective, showing gains OOD relative to the fine-tuned model without losing ID accuracy (top) and gains ID relative to zero-shot without losing OOD accuracy (bottom).

Robustness on additional distribution shifts. Beyond the main five ImageNet derived distribution shifts, WiSE-FT consistently improves robustness on a diverse set of distributions shifts including geographic shifts in satellite imagery and wildlife recognition (WILDS-FMoW, WILDS-iWildCam), reproductions of the popular image classification dataset CIFAR-10 with a distribution shift (CIFAR-10.1 and CIFAR-10.2), and datasets with distribution shift induced by temporal perturbations in videos (ImageNet-Vid-Robust and YTBB-Robust). Concretely, WiSE-FT improves OOD performance by 3.7, 6.5, 2.2, 3.0, 8.3 and 14.7 percentage points respectively, while maintaining or improving ID accuracy. In contrast to the ImageNet distribution shifts, the zero-shot model initially achieves low accuracy on the WILDS distribution shifts, and WiSE-FT provides improvements regardless. More details are included in Appendix C.4.

Additional experiments and analysis. Appendicies C.2 and C.3 demonstrate that WiSE-FT outperforms various baselines including regularization, distillation, and checkpoints saved throughout fine-tuning. Beyond robustness, Appendix C.5 demonstrates that WiSE-FT provides accuracy improvements on ID data, with a focus on the low-data regime. Finally, Section D analyzes the empirical phenomena, finding zero-shot and fine-tuned models are complementary and connecting WiSE-FT to earlier work on the phenomenology of deep learning.

*Although this table considers ImageNet class names, ObjectNet provides alternative class names which can improve the performance of zero-shot CLIP by 2.3 percentage points (Appendix G.4).

References

[1] 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. https://openreview.net/forum?id=OQ08SN70M1V.

[2] Michael A Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. https://arxiv.org/abs/1811.11553.

[3] Anders Andreassen, Yasaman Bahri, Behnam Neyshabur, and Rebecca Roelofs. The evolution of out-of-distribution robustness throughout fine-tuning, 2021. https://arxiv.org/abs/2106.15831.

[4] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL https://proceedings.neurips.cc/paper/2019/file/97af07a14cacba681feacf3012730892-Paper.pdf.

[5] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 1999. https://link.springer.com/article/10.1023/A:1007515423169.

[6] Sara Beery, Arushi Agarwal, Elijah Cole, and Vighnesh Birodkar. The iwildcam 2021 competition dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR) FGVC8 Workshop*, 2021. https://arxiv.org/abs/2105.03494.

[7] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 2018. https://arxiv.org/abs/1712.03141.

[8] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, 2013. https://arxiv.org/abs/1708.06131.

[9] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models, 2021. https://arxiv.org/abs/2108.07258.

[10] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, 2014. https://data.vision.ee.ethz.ch/cvl/datasets_extra/food-101/.

[11] Leo Breiman. Bagging predictors. *Machine learning*, 1996. https://link.springer.com/article/10.1007/BF00058655.

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. https://arxiv.org/abs/2005.14165.

[13] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. https://arxiv.org/abs/1711.07846.

[14] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. https://arxiv.org/abs/1311.3618.

[15] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning (ICML)*, 2019. https://arxiv.org/abs/1902.02918.

[16] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al.

Underspecification presents challenges for credibility in modern machine learning, 2020. https://arxiv.org/abs/2011.03395.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, 2009. https://ieeexplore.ieee.org/document/5206848.

[18] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. https://arxiv.org/abs/2006.06666.

[19] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout, 2017. https://arxiv.org/abs/1708.04552.

[20] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, 2000. https://link.springer.com/chapter/10.1007/3-540-45014-9_1.

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. https://arxiv.org/abs/2010.11929.

[22] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning (ICML)*, 2019. https://arxiv.org/abs/1712.02779.

[23] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. https://arxiv.org/abs/1707.08945.

[24] Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. https://arxiv.org/abs/2010.15110.

[25] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning (ICML)*, 2020. https://arxiv.org/abs/1912.05671.

[26] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1997. https://www.sciencedirect.com/science/article/pii/S002200009791504X.

[27] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*. Springer series in statistics New York, 2001.

[28] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations (ICLR)*, 2018. https://arxiv.org/abs/1811.12231.

[29] Robert Geirhos, Carlos R Medina Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. https://arxiv.org/abs/1808.08750.

[30] Ian J Goodfellow, Oriol Vinyals, and Andrew M Saxe. Qualitatively characterizing neural network optimization problems. In *International Conference on Learning Representations (ICLR)*, 2014. https://arxiv.org/abs/1412.6544.

[31] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017. https://arxiv.org/abs/1706.04599.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. https://arxiv.org/abs/1512.03385.

[33] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *International Conference on Learning Representations (ICLR)*, 2019. https://arxiv.org/abs/1903.12261.

[34] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations (ICLR)*, 2020. https://arxiv.org/abs/1912.02781.

[35] 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. *International Conference on Computer Vision (ICCV)*, 2021. https://arxiv.org/abs/2006.16241.

[36] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. https://arxiv.org/abs/1907.07174.

[37] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Advances in Neural Information Processing Systems (NeurIPS) Deep Learning Workshop*, 2015. https://arxiv.org/abs/1503.02531.

[38] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 1998. https://ieeexplore.ieee.org/document/709601.

[39] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018. https://arxiv.org/abs/1803.05407.

[40] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. https://arxiv.org/abs/1806.07572.

[41] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, 2021. https://arxiv.org/abs/2102.05918.

[42] 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 *Association for Computational Linguistics (ACL)*, 2019. https://arxiv.org/abs/1911.03437.

[43] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences (PNAS)*, 2017. https://arxiv.org/abs/1612.00796.

[44] 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. https://arxiv.org/abs/2012.07421.

[45] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European Conference on Computer Vision (ECCV)*, 2020. https://arxiv.org/abs/1912.11370.

[46] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, 2019. https://arxiv.org/abs/1905.00414.

[47] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. https://arxiv.org/abs/1805.08974.

[48] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *International Conference on Computer Vision (ICCV) Workshops*, 2013. https://ieeexplore.ieee.org/document/6755945.

[49] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[50] Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 2003. https://doi.org/10.1023/A:1022859003006.

[51] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. https://arxiv.org/abs/1612.01474.

[52] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations (ICLR)*, 2018. https://arxiv.org/abs/1804.08838.

[53] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2016. https://arxiv.org/abs/1608.03983.

[54] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. https://openreview.net/forum?id=Bkg6RiCqY7.

[55] Shangyun Lu, Bradley Nott, Aaron Olson, Alberto Todeschini, Hossein Vahabi, Yair Carmon, and Ludwig Schmidt. Harder or different? a closer look at distribution shift in dataset reproduction. In *International Conference on Machine Learning (ICML) Workshop on Uncertainty and Robustness in Deep Learning*, 2020. http://www.gatsby.ucl.ac.uk/~balaji/udl2020/accepted-papers/UDL2020-paper-101.pdf.

[56] Ekdeep Singh Lubana, Puja Trivedi, Danai Koutra, and Robert P. Dick. How do quadratic regularizers prevent catastrophic forgetting: The role of interpolation, 2021. https://arxiv.org/abs/2102.02805.

[57] James Lucas, Juhan Bae, Michael R Zhang, Stanislav Fort, Richard Zemel, and Roger Grosse. Analyzing monotonic linear interpolation in neural network loss landscapes. In *International Conference on Machine Learning (ICML)*, 2021. https://arxiv.org/abs/2104.11044.

[58] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2017. https://arxiv.org/abs/1706.06083.

[59] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 1989. https://www.sciencedirect.com/science/article/pii/S0079742108605368.

[60] Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 2012.

[61] John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. The effect of natural distribution shift on question answering models. In *International Conference on Machine Learning (ICML)*, 2020. https://arxiv.org/abs/2004.14444.

[62] John P 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. https://arxiv.org/abs/2107.04649.

[63] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. https://arxiv.org/abs/1906.02629.

[64] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Deep ensembles for low-data transfer learning, 2020. https://arxiv.org/abs/2010.06866.

[65] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. https://arxiv.org/abs/2008.11687.

[66] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms, 2018. https://arxiv.org/abs/1803.02999.

[67] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. https://arxiv.org/abs/1906.02530.

[68] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. https://arxiv.org/abs/1912.01703.

[69] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 1992. https://epubs.siam.org/doi/abs/10.1137/0330046?journalCode=sjcodc.

[70] Joaquin Quiñonero-Candela, Masashi Sugiyama, Neil D Lawrence, and Anton Schwaighofer. *Dataset shift in machine learning*. Mit Press, 2009.

[71] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners, 2019. https://openai.com/blog/better-language-models/.

[72] 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)*, 2021. https://arxiv.org/abs/2103.00020.

[73] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *International Conference on Machine Learning (ICML)*, 2019. https://arxiv.org/abs/1902.10811.

[74] David Ruppert. Efficient estimations from a slowly convergent robbins-monro process, 1988. https://ecommons.cornell.edu/handle/1813/8664.

[75] Hadi Salman, Greg Yang, Jerry Li, Pengchuan Zhang, Huan Zhang, Ilya Razenshteyn, and Sebastien Bubeck. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. https://arxiv.org/abs/1906.04584.

[76] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. In *European Conference on Computer Vision (ECCV)*, 2020. https://arxiv.org/abs/2008.01392.

[77] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. https://arxiv.org/abs/1904.12843.

[78] Vaishaal Shankar, Achal Dave, Rebecca Roelofs, Deva Ramanan, Benjamin Recht, and Ludwig Schmidt. Do image classifiers generalize across time?, 2019. https://arxiv.org/abs/1906.02168.

[79] Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. Evaluating machine accuracy on imagenet. In *International Conference on Machine Learning (ICLR)*, 2020.

[80] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014. https://arxiv.org/abs/1403.6382.

[81] David B Skalak et al. The sources of increased accuracy for two proposed boosting algorithms. In *American Association for Artificial Intelligence (AAAI), Integrating Multiple Learned Models Workshop*, 1996. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.40.2269&rep=rep1&type=pdf.

[82] Asa Cooper Stickland and Iain Murray. Diverse ensembles improve calibration. In *International Conference on Machine Learning (ICML) Workshop on Uncertainty and Robustness in Deep Learning*, 2020. https://arxiv.org/abs/2007.04206.

[83] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. https://arxiv.org/abs/1912.04838.

[84] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, 2019. https://proceedings.mlr.press/v97/tan19a/tan19a.pdf.

[85] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. https://arxiv.org/abs/2007.00644.

[86] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. https://people.csail.mit.edu/torralba/publications/datasets_cvpr11.pdf.

[87] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations (ICLR)*, 2017. https://arxiv.org/abs/1705.07204.

[88] 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. https://arxiv.org/abs/1905.13549.

[89] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[90] Mitchell Wortsman, Maxwell C Horton, Carlos Guestrin, Ali Farhadi, and Mohammad Rastegari. Learning neural network subspaces. In *International Conference on Machine Learning (ICML)*, 2021. https://arxiv.org/abs/2102.10472.

[91] Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 2016. https://link.springer.com/article/10.1007/s11263-014-0748-y.

[92] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. https://arxiv.org/abs/1911.04252.

[93] LI Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning (ICML)*, 2018. https://arxiv.org/abs/1802.01483.

[94] Chhavi Yadav and Léon Bottou. Cold case: The lost mnist digits. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. https://arxiv.org/abs/1905.10498.

[95] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification, 2019. https://arxiv.org/abs/1905.00546.

[96] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning (ICML)*, 2017. https://arxiv.org/abs/1703.04200.

[97] Michael R Zhang, James Lucas, Geoffrey Hinton, and Jimmy Ba. Lookahead optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. https://arxiv.org/abs/1907.08610.

[98] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text, 2020. https://arxiv.org/abs/2010.00747.

[99] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models, 2021. https://arxiv.org/abs/2109.01134.

[100] 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. `https://arxiv.org/abs/1909.11764`.

A Experimental setup: additional details

A.1 ImageNet distribution shifts

Concretely, we consider: ImageNet-V2 [IN-V2, 73], a reproduction of the ImageNet test set with distribution shift; ImageNet-R [IN-R, 35], renditions (e.g. sculptures, paintings) for 200 ImageNet classes; ImageNet Sketch [IN-Sketch, 88], which contains sketches instead of natural images; ObjectNet [4], a test set of objects in various scenes with 113 classes overlapping with ImageNet; ImageNet-A [IN-A, 36], a test set of natural images misclassified by a ResNet-50 [32] for 200 ImageNet classes.

A.2 Effective robustness and scatter plots

For a number of distribution shifts, accuracy on the standard test set is a reliable predictor of accuracy under distribution shift [85, 62]. In other words, there exists a function $\beta : [0, 1] \to [0, 1]$ such that $\mathsf{Acc}_{\mathcal{D}'}(f)$ approximately equals $\beta(\mathsf{Acc}_{\mathcal{D}}(f))$ for models f trained on the in-distribution train set $\mathcal{S}_{\mathcal{D}}^{\text{tr}}$. Effective robustness [85] is accuracy beyond this baseline, defined formally as $\rho(f) = \mathsf{Acc}_{\mathcal{D}'}(f) - \beta(\mathsf{Acc}_{\mathcal{D}}(f))$.

In the corresponding scatter plots, effective robustness is vertical movement above expected OOD performance (Figure 1, top), and disentangles the effects of ID accuracy changes in OOD robustness interventions. When we say that a model is robust to distribution shift, we mean that effective robustness is positive. Taori et al. [85] observed that no algorithmic robustness intervention consistently achieves substantial effective robustness across the distribution shifts in Figure 2—the first to do so was zero-shot CLIP. Empirically, when applying logit (or probit) axis scaling, models trained on the ID training data approximately lie on a linear trend [85, 62, Figure 1]. As in Taori et al. [85], we apply logit axis scaling and show 95% Clopper-Pearson confidence intervals for the accuracies of select points.

The scatter plots we display also show a wide range of machine learning models from a comprehensive testbed of evaluations [85, 62], including: models trained on $\mathcal{S}_{\mathcal{D}}^{\text{tr}}$ (*standard training*); models trained on additional data and fine-tuned using $\mathcal{S}_{\mathcal{D}}^{\text{tr}}$ (*trained with more data*); and models trained using various *existing robustness interventions*, e.g. special data augmentation [19, 22, 28, 34] or adversarially robust models [58, 15, 75, 77].

A.3 Zero-shot models and CLIP

Zero-shot CLIP models exhibit effective robustness and lie on a qualitatively different linear trend [Figure 1, Figure 13 in 72]. These models are pre-trained using image-caption pairs from the web. Given a set of image-caption pairs $\{(x_1, s_1)..., (x_B, s_B)\}$, CLIP trains an image-encoder g and text-encoder h such that the similarity $\langle g(x_i), h(s_i)\rangle$ is maximized relative to unaligned pairs. CLIP-like models perform zero-shot k-way classification given an image x and class names $C = \{c_1, ..., c_k\}$ by matching x with potential captions. For instance, using caption $s_i =$ "a photo of a $\{c_i\}$" for each class i, the zero-shot model predicts the class via $\arg\max_j \langle g(x), h(s_j)\rangle$. [1] Equivalently, one can construct $\mathbf{W}_{\text{zero-shot}} \in \mathbb{R}^{d\times k}$ with columns $h(s_j)$ and compute outputs $f(x) = g(x)^\top \mathbf{W}_{\text{zero-shot}}$. Unless explicitly mentioned, our experiments use the CLIP model `ViT-L/14@336px`, although all CLIP models are displayed in our scatter plots.

B Motivation and detail for WiSE-FT

This section describes and motivates our proposed method, WiSE-FT, which consists of two steps. First, we fine-tune the zero-shot model on application-specific data. Second, we combine the original zero-shot and fine-tuned models by linearly interpolating between their weights, also referred to as weight-space ensembling.

The zero-shot model excels under distribution shift while standard fine-tuning achieves high ID accuracy. Our motivation is to combine these two models into one that achieves the best of both

[1] For improved accuracy, the embedding of a few candidate captions are averaged, e.g., $s_i^{(1)} =$ "a *photo* of a $\{c_i\}$" and $s_i^{(2)} =$ "a *picture* of a $\{c_i\}$" (referred to as prompt ensembling [72]).

worlds. Weight-space ensembles are a natural choice as they ensemble without extra computational cost. Moreover, previous work has suggested that interpolation in weight space may improve performance when models share part of their optimization trajectory [39, 65].

Step 1: Standard fine-tuning. Let $\mathcal{S}_{\mathcal{D}}^{\text{tr}}$ denote the dataset used for fine-tuning and g denote the image encoder used by CLIP. We are now explicit in writing $g(x, \mathbf{V}_{\text{enc}})$ where x is an input image and $\mathbf{V}_{\text{enc}}$ are the parameters of the encoder g.

Standard fine-tuning considers the model $f(x, \theta) = g\left(x, \mathbf{V}_{\text{enc}}\right)^{\top} \mathbf{W}_{\text{classifier}}$ where $\mathbf{W}_{\text{classifier}} \in \mathbb{R}^{d \times k}$ is the classification head and $\theta = [\mathbf{V}_{\text{enc}}, \mathbf{W}_{\text{classifier}}]$ are the parameters of f. We then solve $\arg\min_{\theta} \left\{ \sum_{(x_i, y_i) \in \mathcal{S}_{\mathcal{D}}^{\text{tr}}} \ell(f(x_i, \theta),\ y_i) + \lambda R(\theta) \right\}$ where ℓ is the cross-entropy loss and R is a regularization term (e.g., weight decay). We consider the two most common variants of fine-tuning: end-to-end, where all values of θ are modified, and fine-tuning only a linear classifier, where $\mathbf{V}_{\text{enc}}$ is fixed at the value learned during pre-training. Additional details provided in Appendices G.2 and G.3.

Step 2: Weight-space ensembling. For a *mixing coefficient* $\alpha \in [0, 1]$, we consider the *weight-space ensemble* between the zero-shot model with parameters θ_0 and the model obtained via standard fine-tuning with parameters θ_1. The predictions of the weight-space ensemble wse are given by

$$\mathsf{wse}(x, \alpha) = f(x, (1 - \alpha) \cdot \theta_0 + \alpha \cdot \theta_1)\ , \tag{1}$$

i.e., we use the element-wise weighted average of the zero-shot and fined-tuned parameters. When fine-tuning only the linear classifier, weight-space ensembling is equivalent to the traditional output-space ensemble [20, 11, 26] $(1 - \alpha) \cdot f(x, \theta_0) + \alpha \cdot f(x, \theta_1)$ since Equation 1 decomposes as $(1 - \alpha) \cdot g(x, \mathbf{V}_{\text{enc}})^{\top} \mathbf{W}_{\text{zero-shot}} + \alpha \cdot g(x, \mathbf{V}_{\text{enc}})^{\top} \mathbf{W}_{\text{classifier}}$.

As neural networks are non-linear with respect to their parameters, ensembling all layers—as we do when end-to-end fine-tuning—typically fails, achieving no better accuracy than a randomly initialized neural network [25]. However, as similarly observed by previous work where part of the optimization trajectory is shared [39, 25, 65], we find that the zero-shot and fine-tuned models are connected by a linear path in weight-space along which accuracy remains high (explored further in Section D.2).

Remarkably, as we show in Section 3, WiSE-FT boosts OOD accuracy relative to the fine-tuned model without decreasing ID performance. These improvements come without any additional computational cost as a single set of weights is used. We provide PyTorch pseudocode for WiSE-FT in Appendix F.

	Fine-tuning	IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A	Avg
OOD improvement relative to fine-tuned without decreasing ID	Linear classifier	1.1	4.2	2.7	2.6	2.9	2.7
	End-to-end	2.6	10.5	7.1	8.8	15.6	8.9
ID improvement relative to zero-shot without decreasing OOD	Linear classifier	8.8	1.7	8.7	7.2	2.4	5.8
	End-to-end	10.6	1.9	10.6	8.8	2.7	6.9

Table 2: Compared to the fine-tuned model, WiSE-FT improves OOD accuracy without reducing ID accuracy. Compared to the zero-shot model, WiSE-FT improves ID accuracy without reducing OOD accuracy. All numbers are percentage point improvements. Top and bottom respectively capture vertical movement above the fine-tuned model (i.e., improved effective robustness) and horizontal movement to the right of the zero-shot model in the associated scatter plots.

C Additional Results

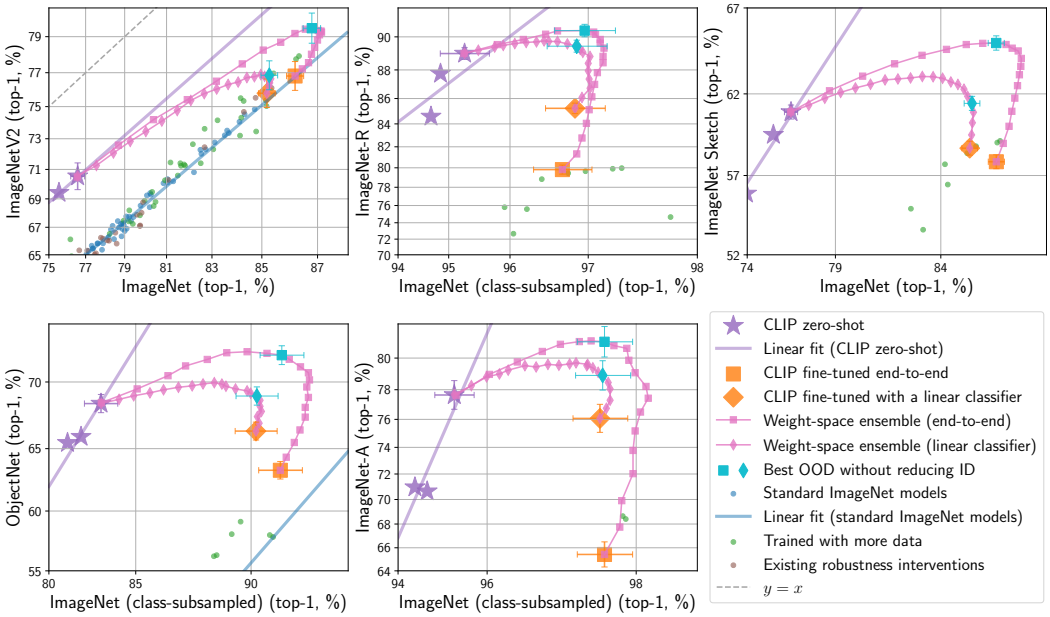

Figure 3: A per-dataset breakdown of the key experimental results (Figure 1). WiSE-FT improves ID and OOD accuracy on ImageNet and five derived distribution shifts. Standard ImageNet models, models trained with more data, and existing robustness interventions are from Taori et al. [85].

C.1 Breakdown of experimental findings on ImageNet and five derived distribution shifts

In contrast to Figure 1, where our key experimental results for ImageNet and five derived distribution shifts are averaged, we now display the results separately for each distribution shift. Results are provided in Figures 3. WiSE-FT improves OOD accuracy while improving or maintaining ID accuracy for each individual distribution shift.

C.2 Comparison to alternative methods (fine-tuning end-to-end)

In search of a fine-tuning method that preserves robustness, we explore a variety of alternatives. Many exhibit a concave trend in effective robustness plots, although WiSE-FT offers the best results overall. Figure 4 compares three alternatives for end-to-end fine-tuning: *output-space ensembles* combine the outputs assigned by the zero-shot and fine-tuned models, i.e., $(1-\alpha) \cdot f(x, \theta_0) + \alpha \cdot f(x, \theta_1)$ for $\alpha \in [0, 1]$; *eval along trajectory* evaluates the model at various iterations throughout training (at each iteration that is a power of two in the first epoch and after every subsequent epoch); *regularize to zero-shot* appends the quadratic regularizer $\lambda \|\theta - \theta_0\|_2^2$ to the fine-tuning objective, where θ are the parameters being learned and θ_0 are the zero-shot parameters.

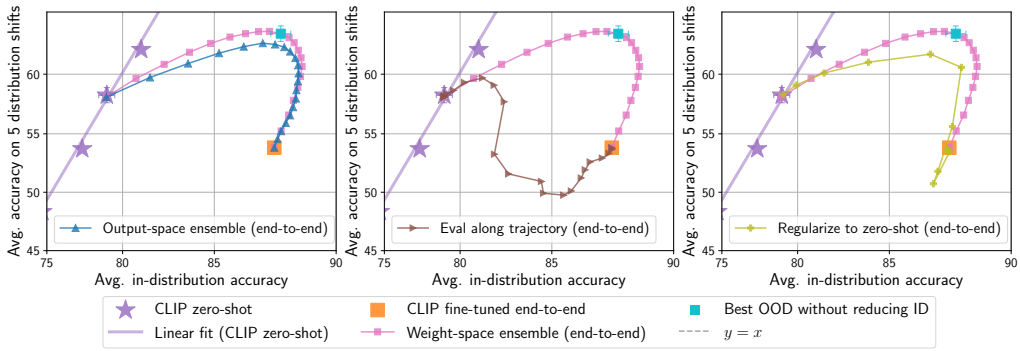

Figure 4: When fine-tuning end-to-end, WiSE-FT outperforms output-space ensembles, intermediate checkpoints, and quadratic regularization (results shown for `ViT-B/16` CLIP).

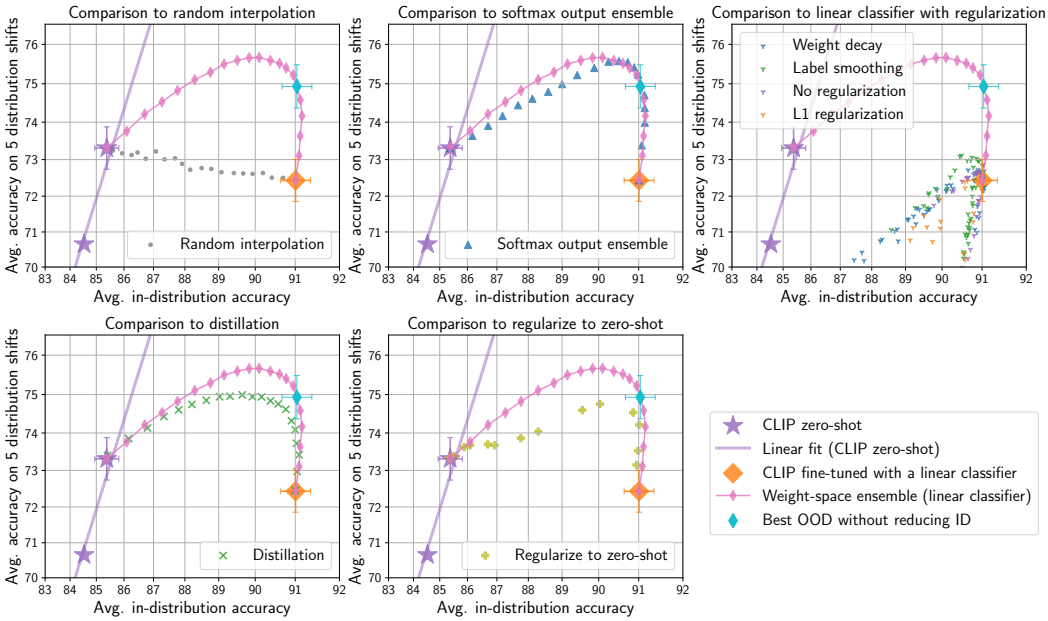

Figure 5: Comparing the relative ID and OOD accuracy of weight-space ensembling with the alternatives described in Section C.3. Many methods follow a concave trend, though weight-space ensembling provides the best performance overall.

C.3 Comparison to alternative methods (linear classifier fine-tuning)

We now extend Appendix C.2 and compare WiSE-FT to additional methods of fine-tuning. Unlike Appendix C.2, our comparisons now focus on fine-tuning only a linear classifier, allowing comprehensive experimentation. Many exhibit a concave trend in effective robustness plots, although WiSE-FT offers the best results overall (Figure 5).

Random interpolation. This method uses either the zero-shot or fine-tuned linear classifier depending on a (biased) coin flip. For hyperparameter $\alpha \in [0, 1]$ outputs are computed as $(1 - \xi) \cdot f(x, \theta_0) + \xi \cdot f(x, \theta_1)$ where ξ is a Bernoulli(α) random variable. For this method and all others with a hyperparameter $\alpha \in [0, 1]$ we evaluate models for $\alpha \in \{0, 0.05, 0.1, ..., 1\}$.

Ensembling softmax outputs. Instead of ensembling in weight space, this method combines softmax probabilities assigned by the zero-shot and fine-tuned linear classifier. Concretely, for hyperparameter $\alpha \in [0, 1]$ outputs are computed as $(1-\alpha)\cdot\mathsf{softmax}(f(x, \theta_0))+\alpha\cdot\mathsf{softmax}(f(x, \theta_1))$. This method performs comparably to weight-space ensembling but requires slightly more compute.

Linear classifier with various regularizers. We explore fine-tuning linear classifiers with four regularization strategies: no regularization, weight decay, L1 regularization, and label smoothing [63]. Linear-classifiers are trained with mini-batch optimization, using the AdamW optimizer [54, 68]

	IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A	Avg
Weight-space ensemble	**1.1**	**4.2**	2.7	2.6	**2.9**	**2.7**
Output-space ensemble	0.8	4.1	**3.1**	**2.7**	2.7	**2.7**
Distillation	0.4	2.7	1.2	1.0	1.7	1.4
Regularize to zero-shot	0.0	3.6	0.0	1.8	0.0	1.1

Table 3: OOD accuracy gain (percentage points) without any loss in ID accuracy relative to the fine-tuned linear classifier for the methods described in Section C.3.

with a cosine-annealing learning rate schedule [53]. This method is significantly faster and less memory-intensive than the L-BFGS implementation used by Radford et al. [72] at ImageNet scale with similar accuracy. Additional details on hyperparameters and more analyses are provided in Appendix G.3.

Distillation. Network distillation [37] trains one network to match the outputs of another. We use this technique to fine-tune while matching the outputs of the zero-shot model, in an attempt to boost out-of-distribution performance. For a hyperparameter $\alpha \in [0, 1]$ and cross-entropy loss ℓ we fine-tune θ according to the minimization objective

$$\sum_{(x_i, y_i) \in \mathcal{S}_{\mathcal{D}}^{\text{tr}}} (1 - \alpha) \cdot \ell(f(x_i, \theta), y_i) + \alpha \cdot \ell(f(x_i, \theta), f(x_i, \theta_0)) \ . \tag{2}$$

Regularization towards zero-shot. We train a linear classifier with an additional regularization term which penalizes movement from the zero-shot classifier's weights. For a hyperparameter $\lambda \in \{1 \cdot 10^{-8}, 5 \cdot 10^{-8}, 1 \cdot 10^{7}, ..., 5 \cdot 10^{2}\}$ we add the regularization term $\lambda \left\| \mathbf{W} - \mathbf{W}_{\text{zero-shot}} \right\|_F^2$ where $\mathbf{W}$ is the linear classifier being fine-tuned. In most cases this method performs slightly worse than distillation.

Figures 24-28 in Appendix J provide a breakdown of Figure 5 for each dataset. Table 3 offers an alternative perspective, showing the amount of out-of-distribution accuracy which can be gained without reducing in-distribution performance for each method.

Figure 6 demonstrates that WiSE-FT achieves better OOD and ID accuracy than the recently proposed CoOp method [99] on ImageNet and four derived distribution shifts. Instead of fine-tuning network parameters, CoOp instead learns continuous embedding for the language prompts. We note that CoOp and WiSE-FT could be used in conjunction in future work. We compare with the `ViT-B/16` section in Table 7 of Zhou et al. [99]. For comparison we use the same CLIP model as CoOp and also train only on 16 images per class. Finally, we note the reported zero-shot numbers of Zhou et al. [99] are slightly lower than Radford et al. [72] as Zhou et al. [99] do not use prompt ensembling.

C.4 Robustness on additional distribution shifts

Figure 7 displays the effective robustness scatter plots for the six additional distribution shifts discussed in Section 3. On each, WiSE-FT improves OOD accuracy without any loss in ID performance.

Concretely, we consider: (i) ImageNet-Vid-Robust and YTBB-Robust, datasets with distribution shift induced by temporal perturbations in videos [78]; (ii) CIFAR-10.1 [73] and CIFAR-10.2 [55], reproductions of the popular image classification dataset CIFAR-10 [49] with a distribution shift; (iii) WILDS-FMoW, a satellite image recognition task where the test set has a geographic and temporal distribution shift [44, 13]; (iv) WILDS-iWildCam, a wildlife recognition task where the test set has a geographic distribution shift [44, 6].

C.5 In-distribution accuracy improvements

In addition to robustness to distribution shift, Table 4 and Figure 8 demonstrates that WiSE-FT is able to improve ID performance on a number of datasets. When fine-tuning end-to-end on ImageNet, CIFAR-10, CIFAR-100, Describable Textures, Food-101, SUN397, Stanford Cars, relative error is reduced by 6 to 62%. This is surprising as standard fine-tuning optimizes for low ID error.

In many application-specific scenarios, only a small amount of data is available for fine-tuning. Accordingly, we examine the performance of WiSE-FT when only k examples per class are used for fine-tuning on the seven aforementioned datasets ($k = \{1, 5, 10, 25, 50\}$). In contrast with Figure 8,

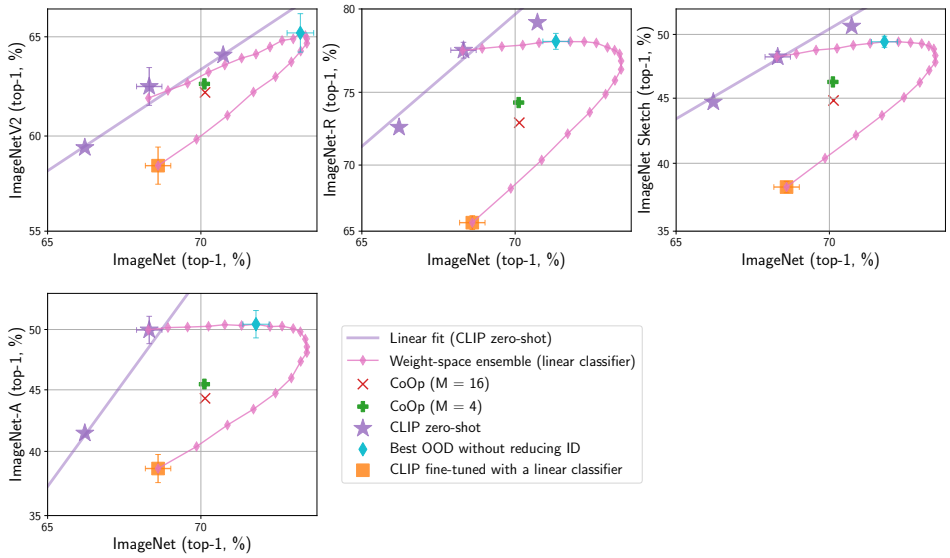

Figure 6: Comparing WiSE-FT with CoOp [99]. Both methods fine-tune the `ViT-B/16` CLIP model on 16 examples per class of ImageNet.

	ImageNet	CIFAR10	CIFAR100	StanfordCars	DTD	SUN397	Food101
Accuracy gain (percentage points)	0.9	0.8	1.2	2.0	3.3	2.6	1.8
Relative error reduction (%)	6.6	61.6	15.3	23.6	18.2	13.3	32.6

Table 4: In addition to robustness, WiSE-FT with optimal α is able to improve in-distribution performance on a number of datasets compared to standard fine-tuning.

we now fine-tune only the linear classifier allowing for comprehensive experiments. When five examples per class are used for fine-tuning, WiSE-FT improves in-distribution performance by 0.3 to 6.1 percentage points compared to the best of the zero-shot and fine-tuned models. Average results are shown in Figure 9, while Figures 10, 11 provide a breakdown for all datasets.

C.6 Robustness across scales of pre-training compute

The strong correlation between standard test accuracy and accuracy under distribution shift holds from low to high performing models. This offers the opportunity to explore robustness for smaller, easy to run models. Our exploration began with the lowest accuracy CLIP models and similar trends held at scale. Figure 12 shows improved out-of-distribution accuracy with minimal loss in-distribution across orders of magnitude of pre-training compute with WiSE-FT when fine-tuning only a linear classifier. Moreover, Figure 13 we recreate the experimental results for ImageNet and five associated distribtuion shifts with a smaller CLIP `ViT-B/16` model, finding similar trends.

C.7 Ensembling zero-shot CLIP with independently trained models

So far we have shown that a zero-shot model can be used to improve out-of-distribution performance of the derived fine-tuned model. Here, we investigate whether this improvement is specific to the fine-tuned model. On the contrary, we find that ensembling with robust models can improve out-of-distribution accuracy of *independently trained models*. Note that in the general case where the models being ensembled have different architectures, we are unable to perform weight-space ensembling; instead, we ensemble the outputs of each model. This increases the computational cost of inference, in contrast to the results shown in Section 3.

Concretely, we ensemble zero-shot CLIP with two Noisy Student EfficientNet models [92, 84]: (i) EfficientNet-B6 (Figure 14, left), with in-distribution performance comparable to the end-to-end fine-tuned CLIP model; and (ii) EfficientNet-L2 (Figure 14, right), the strongest model available on PyTorch ImageNet Models [89]. In both cases, we observe substantial improvements from

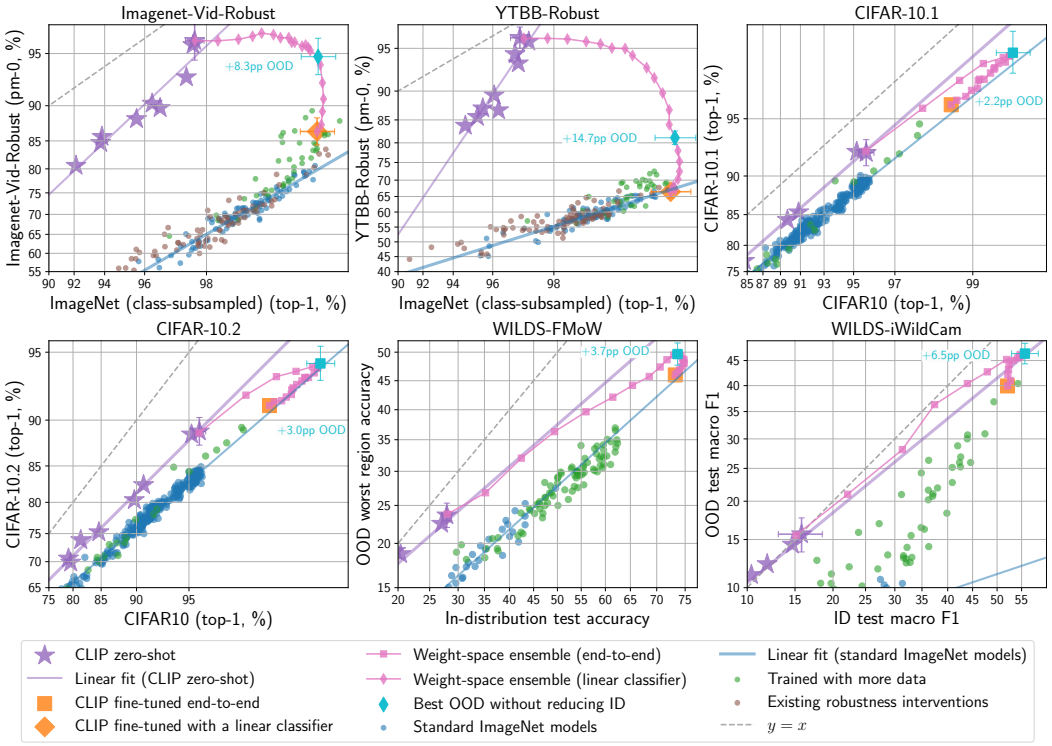

Figure 7: WiSE-FT improves OOD accuracy while maintaining or improving ID accuracy on ImageNet-Vid-Robust, YTBB-Robust [78], CIFAR-10.1 [73], CIFAR-10.2 [55], WILDS-FMoW [44, 13], and WILDS-iWildCam [44, 6]. Numbers reported are percentage point improvements OOD without any loss in ID accuracy compared to standard fine-tuning.

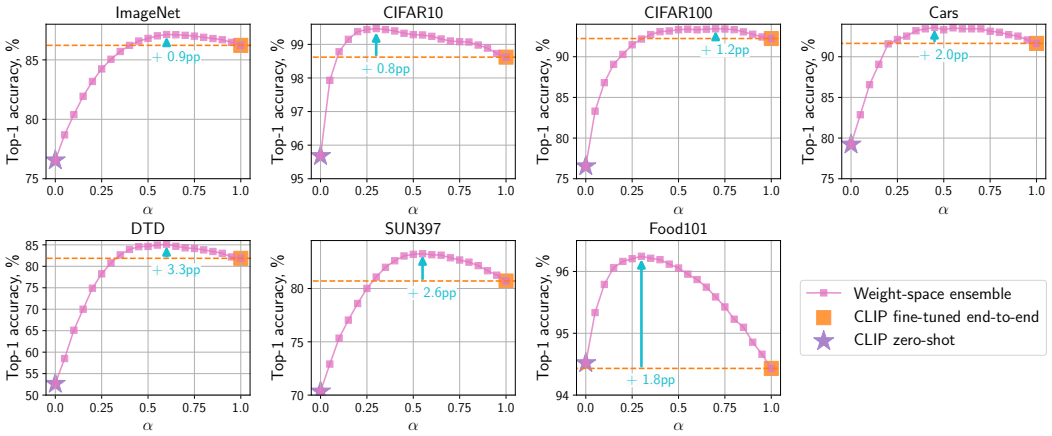

Figure 8: The ID accuracy of WiSE-FT (end-to-end) with mixing coefficient α on ImageNet and a number of datasets considered by Kornblith et al. [47]: CIFAR-10, CIFAR-100 [49], Describable Textures [14], Food-101 [10], SUN397 [91], and Stanford Cars [48].

ensembling—13.6 pp and 6.9 pp in average out-of-domain accuracy without reducing in-distribution performance. Further result are sown in table 5.

D Discussion

This section further analyzes the empirical phenomena we observed so far. We begin with the case where only the final linear layer is fine-tuned and predictions from the weight-space ensemble can be

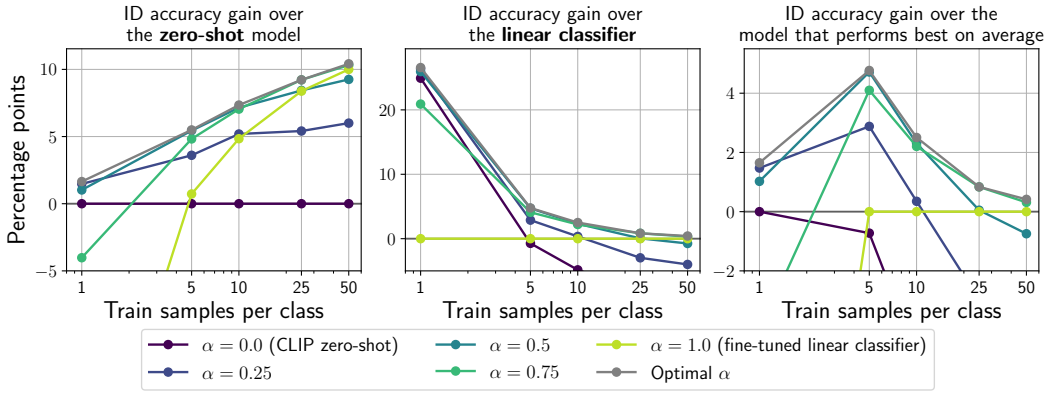

Figure 9: WiSE-FT can improve in-distribution accuracy over the linear classifier and zero-shot model in the low data regime. On the x-axis we consider $k = \{1, 5, 10, 25, 50\}$ examples per class for fine-tuning. On the y-axis we display in-distribution accuracy improvements of WiSE-FT averaged over seven datasets [17, 49, 14, 10, 91, 48]. For $k = 1$, the zero-shot model outperforms the fine-tuned linear classifier, and ensembles closer to the zero-shot model (small α) yield high performance. When more data is available, the reverse is true, and higher values of α improve in-distribution performance. Appendix J, displays a breakdown for all datasets.

		OOD datasets					Avg	Avg
	IN (ID)	IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A	OOD	ID,OOD
CLIP								
End-to-end fine-tuned	86.2	76.8	79.8	57.9	63.3	65.4	68.6	77.4
WiSE-FT, α=0.75	87.0	78.8	86.1	62.5	68.1	75.2	74.1	80.5
WiSE-FT, α=0.4	86.2	79.2	89.9	**65.0**	71.9	80.7	77.3	81.8
WiSE-FT, optimal α	87.1	79.5	90.3	**65.0**	72.1	81.0	77.6	82.3
NS EfficientNet-B6								
No ensemble	86.5	77.7	65.6	47.8	58.3	62.3	62.3	74.4
OSE, α=0.75	87.0	78.8	86.4	56.7	66.5	75.9	72.9	80.0
OSE, α=0.4	84.3	77.2	89.5	63.8	69.7	79.0	75.8	80.0
OSE, optimal α	87.1	79.3	89.7	63.8	69.7	79.3	76.4	81.8
NS EfficientNet-L2								
No ensemble	88.3	80.8	74.6	47.6	69.8	84.7	71.5	79.9
OSE, α=0.75	**88.6**	81.6	88.0	53.4	72.2	**87.1**	76.5	82.5
OSE, α=0.4	85.2	78.5	**90.5**	63.9	72.6	86.0	78.3	81.8
OSE, optimal α	**88.6**	**81.7**	**90.5**	63.9	**73.1**	**87.1**	**79.3**	**83.9**

Table 5: Accuracy of various independently trained models ensembled with CLIP on ImageNet and derived distribution shifts. OSE denotes output-space ensembling. *Avg OOD* displays the mean performance among the five out-of-distribution datasets, while *Avg ID,OOD* shows the average of ImageNet (ID) and Avg OOD.

factored into the outputs of the zero-shot and fine-tuned model. Next, we connect our observations regarding end-to-end fine-tuning with earlier work on the phenomenology of deep learning.

D.1 Zero-shot and fine-tuned models are complementary

In this section, we find that the zero-shot and fine-tuned models have diverse predictions, both ID and OOD. Moreover, while the fine-tuned model is more confident ID, the reverse is true OOD.

Zero-shot and fine-tuned models are diverse. In certain cases, ensemble accuracy is correlated with diversity among the constituents [50]. If two models make coincident mistakes, so will their ensemble, and no benefit will be gained from combining them. Here, we explore two measures of diversity: *Prediction Diversity,* which measures the fraction of examples for which two classifiers disagree but one is correct; and *Centered Kernel Alignment Complement,* the complement of CKA [46]. Additional diversity measures and more details are provided in Appendix H. In Figure 15 (left), we show that the zero-shot and fine-tuned models are diverse both ID and OOD, despite sharing the

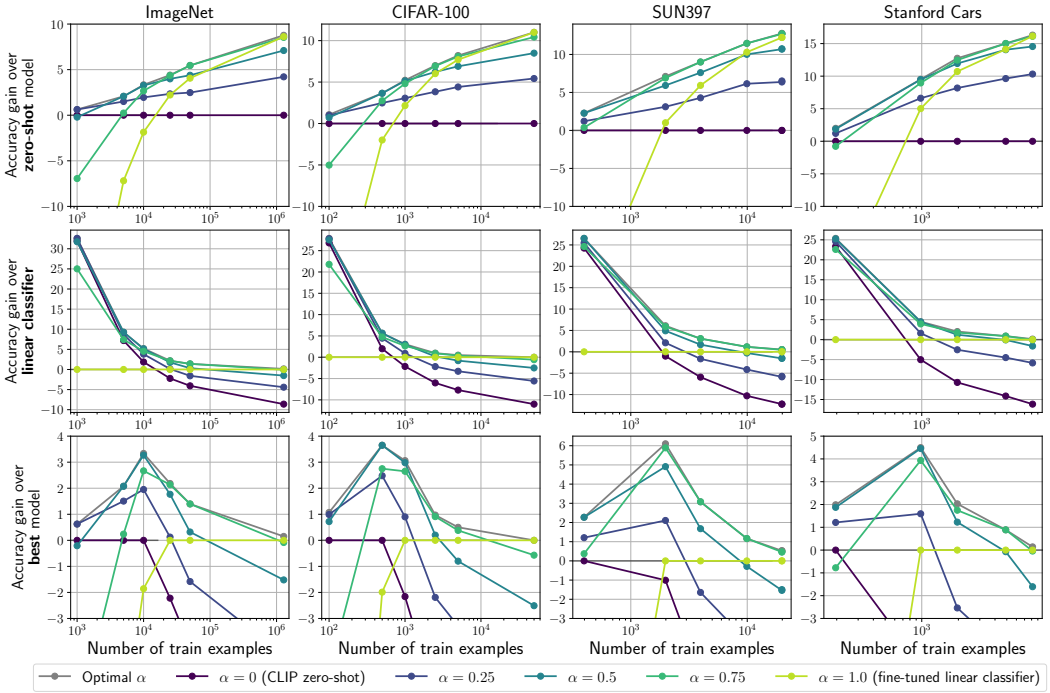

Figure 10: WiSE-FT improves in-distribution accuracy over the linear classifier and zero-shot model in the low data regime. On the x-axis we consider $k = \{1, 5, 10, 25, 50\}$ examples per class and the full training set. On the y-axis we consider the in-distribution accuracy improvement of WiSE-FT over the **(top)** zero-shot model, **(middle)** fine-tuned linear classifier, and **(bottom)** best of the zero-shot and fine-tuned linear classifier.

same backbone. As a point of comparison, we include average diversity measures between two linear classifiers fine-tuned with random splits on half of ImageNet,[2] denoted in orange in Figure 15.

Models are more confident where they excel. In order for the ensemble model to be effective, it should leverage each model's expertise based on which distribution the data is from. Here, we empirically show that this occurs on a number of datasets we consider. First, we examine the cases where the models being ensembled disagree. We say the zero-shot model *overrides* the fine-tuned model if their predictions disagree and the zero-shot prediction matches that of the weight-space ensemble. Similarly, if models disagree and the linear classifier prediction matches the ensemble, we say the zero-shot is *overridden*. Figure 15 (middle) shows the fraction of samples where the zero-shot model overrides and is overridden by the fine-tuned linear classifier for α=0.5. Other than ImageNetV2, which was collected to closely reproduce ImageNet, the zero-shot model overrides the linear classifier more than it is overridden on the OOD datasets. Additionally, we are interested in measuring model confidence. Recall that we are ensembling quantities before a softmax is applied, so we avoid criteria that use probability vectors, e.g., Guo et al. [31]. Instead, we consider the margin δ between the largest and second largest output of each classifier. Figure 15 (right) shows that the zero-shot model is more confident in its predictions for OOD datasets, while the reverse is true ID.

D.2 An error landscape perspective

Our discussion now focuses on the empirical phenomenon we observe when weight-space ensembling all layers in the network. Specifically, this section formalizes our observations and details related phenomena. Recall that the weight-space ensemble of θ_0 and θ_1 is given by $f(x, (1-\alpha)\cdot\theta_0 + \alpha\cdot\theta_1)$ (Equation 1). We begin with a natural generalization of *linear mode connectivity* [25] to the setting where accuracy of the endpoints may differ substantially. For distribution $\mathcal{D}$ and model f, let $\mathsf{Acc}_{\mathcal{D},f}(\theta)$ denote the expected accuracy of f evaluated with parameters θ on distribution $\mathcal{D}$.

[2]Two linear classifiers fine-tuned on the same data converge to similar solutions, resulting in negligible diversity. As a stronger baseline, we fine-tune classifiers on different subsets of ImageNet, with half of the data.

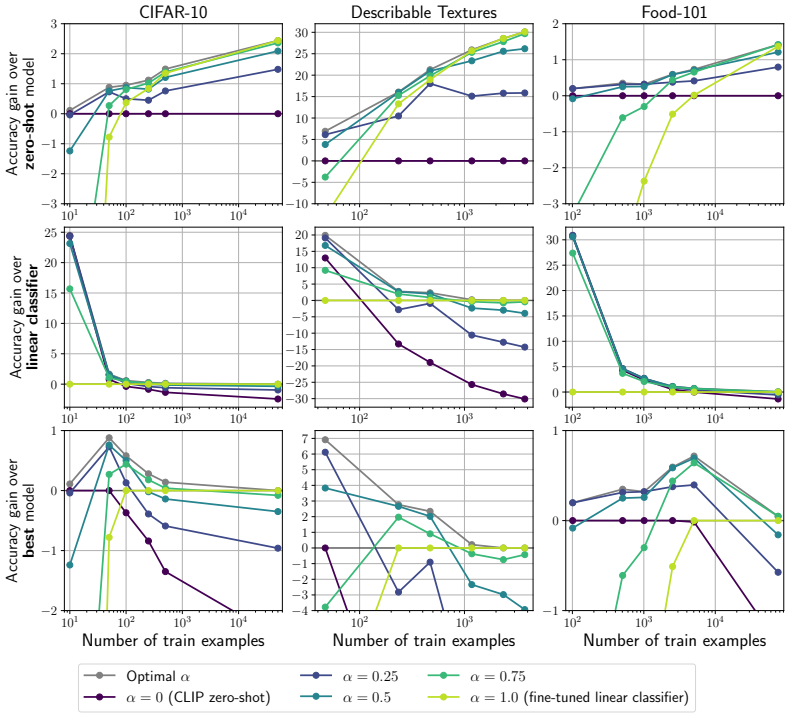

Figure 11: WiSE-FT improves in-distribution accuracy over the linear classifier and zero-shot model in the low data regime. On the x-axis we consider $k = \{1, 5, 10, 25, 50\}$ examples per class and the full training set. On the y-axis we consider the in-distribution accuracy improvement of WiSE-FT over the **(top)** zero-shot model, **(middle)** fine-tuned linear classifier, and **(bottom)** best of the zero-shot and fine-tuned linear classifier.

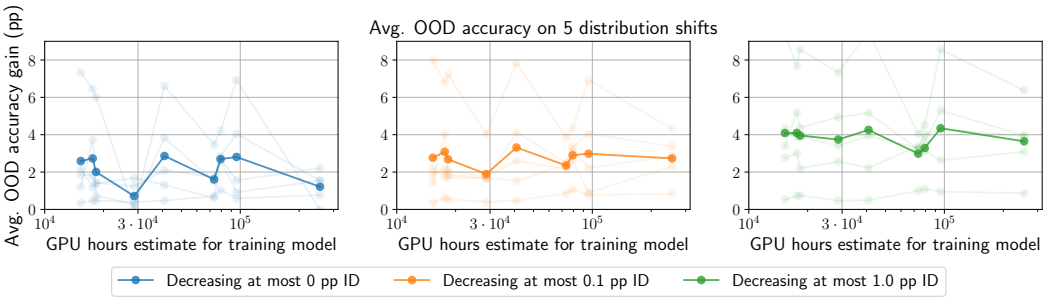

Figure 12: WiSE-FT provides benefits for all CLIP models. Accuracy can be improved out-of-distribution relative to the linear classifier with less than $\epsilon \in \{0, 0.1, 1\}$ percentage points (pp) loss in-distribution across orders of magnitude of training compute. The CLIP model `RN50x64` requires the most GPU hours to train.

Definition 1: Parameters θ_0 and θ_1 exhibit *linear mode connectivity* with respect to f and $\mathcal{D}$ if, for all $\alpha \in [0, 1]$, $\mathsf{Acc}_{\mathcal{D},f}((1-\alpha)\cdot\theta_0 + \alpha\cdot\theta_1) \;\geq\; (1-\alpha)\cdot\mathsf{Acc}_{\mathcal{D},f}(\theta_0) + \alpha\cdot\mathsf{Acc}_{\mathcal{D},f}(\theta_1)$. Note that when $\mathsf{Acc}_{\mathcal{D},f}(\theta_0) = \mathsf{Acc}_{\mathcal{D},f}(\theta_1)$, this is equivalent to the original definition of Frankle et al. [25].[3]

Observation 1: As illustrated in Figure 16, we observe linear mode connectivity on ImageNet and the five associated distribution shifts we consider (Section 2).

To assist in contextualizing Observation 1, we review related phenomena. Neural networks are nonlinear, and hence weight-space ensembles only achieve good performance in exceptional cases—

[3]The original definition used $\gtrapprox \frac{1}{2}(\mathsf{Acc}_{\mathcal{D},f}(\theta_0) + \mathsf{Acc}_{\mathcal{D},f}(\theta_1))$ and so Definition 1 is not a strict generalization. This original definition is less applicable when endpoint accuracy differs substantially.

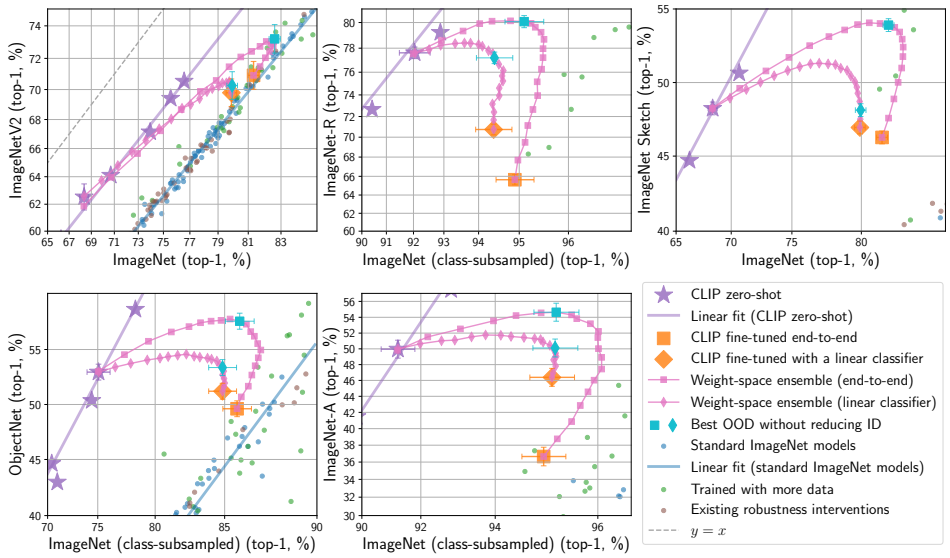

Figure 13: WiSE-FT improves ID and OOD accuracy across a number of distribution shifts with a smaller CLIP `ViT-B/16` model.

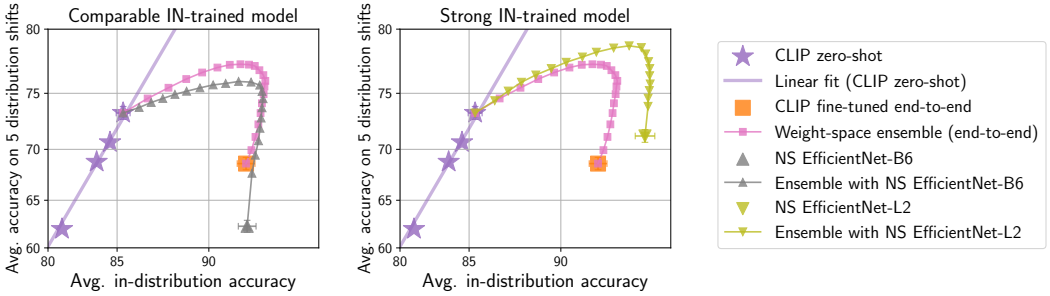

Figure 14: **Ensembling with a zero-shot model improves the out-of-distribution performance of an independently trained model**. **(Left)** Output-space ensembling with an independently trained model (NoisyStudent EfficientNet-B6) with comparable in-distribution performance to the end-to-end fine-tuned model. **(Right)** Output-space ensembling with an independently trained model with strong in-distribution performance (NoisyStudent EfficientNet-L2). Results averaged over the five distribution shifts as in Figure 1.

ensembling two randomly initialized networks in weight-space achieves no better accuracy than a random classifier [25]. Linear mode connectivity has been observed by Frankle et al. [25], Izmailov et al. [39] when part of the training trajectory is shared, and by Neyshabur et al. [65] when two models are fine-tuned with a shared initialization. In particular, the observations of Neyshabur et al. [65] may elucidate why weight-space ensembles attain high accuracy in the setting we consider, as they suggest that fine-tuning remains in a region where solutions are connected by a linear path along which error remains low. Instead of considering the weight-space ensemble of two fine-tuned models, we consider the weight-space ensemble of the *pre-trained* and fine-tuned models. This is only possible for a pre-trained model capable of zero-shot inference, such as CLIP.

Observation 2: As illustrated by Figure 16, on ImageNet and the five associated distribution shifts we consider, weight-space ensembling (end-to-end) may outperform both the zero-shot and fine-tuned models, i.e., there exists an α for which $\mathsf{Acc}_{\mathcal{D},f}\left((1-\alpha)\cdot\theta_0+\alpha\cdot\theta_1\right) \geq \max\left\{\mathsf{Acc}_{\mathcal{D},f}\left(\theta_0\right),\ \mathsf{Acc}_{\mathcal{D},f}\left(\theta_1\right)\right\}$.

We are not the first to observe that when interpolating between models with linear mode connectivity, the accuracy of models along the path may exceed that of either endpoint [39, 65, 90]. Neyshabur et al. [65] conjecture that interpolation could produce solutions closer to the true center of a basin. This

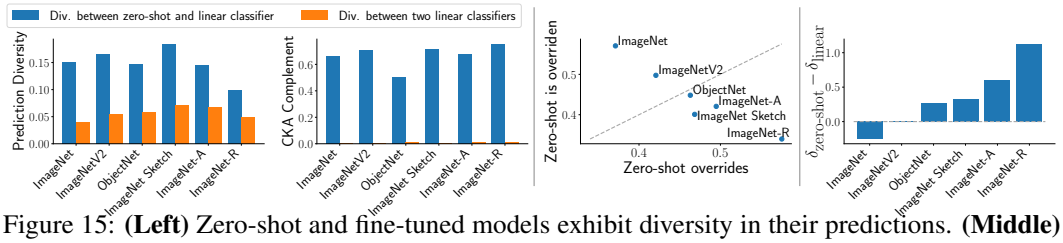

Figure 15: **(Left)** Zero-shot and fine-tuned models exhibit diversity in their predictions. **(Middle)** On most OOD datasets, the zero-shot model overrides the linear classifier more than it is overridden. The reverse is true for ImageNet (in-distribution). **(Right)** Similarly, zero-shot models are more confident on OOD datasets, while the reverse is true in-distribution. The margin δ_f measures the average difference between the largest and second largest unormalized output for classifier f.

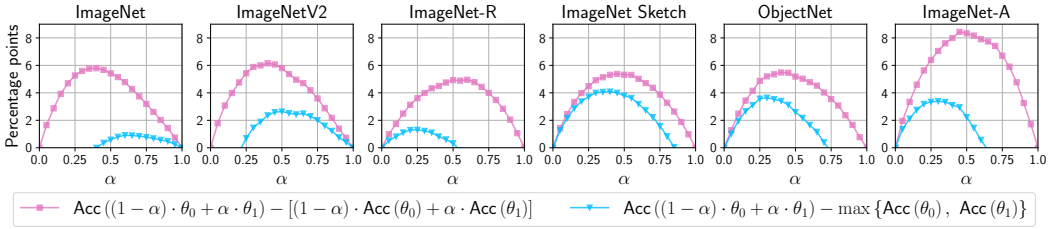

Figure 16: The zero-shot and fine-tuned models exhibit *linear mode connectivity* [25] on ImageNet and the main distribution shifts we consider (Observation 1). Moreover, there exists an α for which WiSE-FT outperforms both the zero-shot and fine-tuned models (Observation 2).

intuition applied in our setting is schematized in Appendix J (Figure 29). In contrast to Neyshabur et al. [65], we interpolate between models which observe different data.

E Related work

Robustness. Understanding how models perform under distribution shift remains an important goal, as real world models may encounter data from new environments [70, 86]. Previous work has studied model behavior under synthetic [33, 87, 58, 29, 23, 2] and natural distribution shift [35, 44, 88, 4, 36]. Interventions used for synthetic shifts do not typically provide robustness to many natural distribution shifts [85]. In contrast, accuracy on the standard test set is a reliable predictor for accuracy under distribution shift [94, 61, 85, 83, 62].

Pre-training and transfer learning. Pre-training on large amounts of data is a powerful technique for building high-performing machine learning systems [80, 21, 45, 95, 71, 12]. One increasingly popular class of vision models are those pre-trained with auxiliary language supervision, which can be used for zero-shot inference [18, 76, 98, 72, 41]. When pre-trained models are adapted to a specific distribution through standard fine-tuning, effective robustness deteriorates at convergence [3]. In natural language processing, previous work proposed stable fine-tuning methods that incur computational overhead [42, 100], alleviating problems such as representational collapse [1]. More generally, a variety of methods have attempted to mitigate catastrophic forgetting in neural networks [59]. Kirkpatrick et al. [43], Zenke et al. [96] explored weighted quadratic regularization for sequential learning. Xuhong et al. [93] showed that for fine-tuning the simple quadratic regularization explored in Section 3 performs best, while Lubana et al. [56] explored the connection between quadratic regularization and interpolation. Andreassen et al. [3] found that many approaches from continual learning do not provide robustness to multiple natural distribution shifts.

Traditional (output-space) ensembles. Traditional ensemble methods, which we refer to as output-space ensembles, combine the predictions (outputs) of many classifiers [20, 5, 11, 27, 51, 26]. Typically, output-space ensembles outperform individual classifiers and provide uncertainty estimates under distribution shift that are more callibrated than baselines [51, 67, 82]. In contrast to these works, we consider the ensemble of two models which have observed different data. Output-space ensembles require more computational resources as they require a separate pass through each model. Compared to an ensemble of 15 models trained on the same dataset, Mustafa et al. [64] find an OOD improvement of 0.8–1.6 pp (on ImageNetV2, ImageNet-R, ObjectNet, and ImageNet-A) by ensembling a similar number of models pre-trained on different datasets. In contrast, we see an improvement of 2–15 pp from ensembling two models. Moreover, as we ensemble in weight-space, no extra compute is required compared to a single model.

Weight-space ensembles. Weight-space ensembles linearly interpolate between the weights of different models [25, 57, 30]. Izmailov et al. [39] average checkpoints saved throughout training for improved performance. Indeed, averaging the weights along the training trajectory is a central method in optimization [74, 69, 66]. For instance, Zhang et al. [97] propose optimizing with a set of fast and slow weights, where every k steps, these two sets of weights are averaged and a new trajectory begins. Here, we revisit these techniques from a distributional robustness perspective and consider the weight-space ensemble of models which have observed different data.

F Pseudocode for WiSE-FT

Algorithm 1 Pytorch pseudocode for WiSE-FT

```
def wse(model, zeroshot_checkpoint, finetuned_checkpoint, alpha):
    # load state dicts from checkpoints
    theta_0 = torch.load(zeroshot_checkpoint)["state_dict"]
    theta_1 = torch.load(finetuned_checkpoint)["state_dict"]

    # make sure checkpoints are compatible
    assert set(theta_0.keys()) == set(theta_1.keys())

    # interpolate between all weights in the checkpoints
    theta = {
        key: (1-alpha) * theta_0[key] + alpha * theta_1[key]
        for key in theta_0.keys()
    }

    # update the model (in-place) according to the new weights
    model.load_state_dict(theta)

def wise_ft(model, dataset, zeroshot_checkpoint, alpha, hparams):
    # load the zero-shot weights
    theta_0 = torch.load(zeroshot_checkpoint)["state_dict"]
    model.load_state_dict(theta_0)

    # standard fine-tuning
    finetuned_checkpoint = finetune(model, dataset, hparams)

    # perform weight-space ensembling (in-place)
    wse(model, zeroshot_checkpoint, finetuned_checkpoint, alpha)
```

G Experimental details

G.1 CLIP Zero-shot

This section extends Section 2 with more details on inference with the CLIP zero-shot model. First, in all settings we use the CLIP model `ViT-L/14@336px` except for: (i) on CIFAR-10 and CIFAR-100 we find that `ViT-L/14` performs slightly better than `ViT-L/14@336px`. (ii) In Figures 4 and 13 where we use `ViT-B/16`. Second, CLIP learns a temperature parameter which is factored into the learned weight matrix $\mathbf{W}_{\text{zero-shot}}$ described in Section 2. Finally, to construct $\mathbf{W}_{\text{zero-shot}}$ we ensemble the 80 prompts provided by CLIP at `https://github.com/openai/CLIP`. However, we manually engineer prompts for five datasets: WILDS-FMoW, WILDS-iWildCam, Stanford Cars, Describable Textures and Food-101, which are found in the code.

G.2 End-to-end fine-tuning

Two important experimental details for fine-tuning are as follows:

- We initialize the final classification layer with the zero-shot classifier used by CLIP. This includes the temperature parameter, which is no longer decoupled during fine-tuning.
- As the zero-shot classifier expects the outputs of the image-encoder g to be normalized, we continue to normalize the outputs of g during fine-tuning.

When fine-tuning end-to-end we use the AdamW optimizer [54, 68] and choose the largest batch size such that the model fits into 8 GPUs (512 for `ViT-B/16`). We use the default PyTorch AdamW hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay of 0.1 and a cosine-annealing learning rate schedule [53] with 500 warm-up steps. We use a learning rate of 3×10^{-5}, gradient clipping at global norm 1 and fine-tune for a total of 10 epochs. We use the same data augmentations as [72], randomly cropping a square from resized images with the largest dimension being 336 pixels for `ViT-L/14@336px` and 224 for the remaining models.

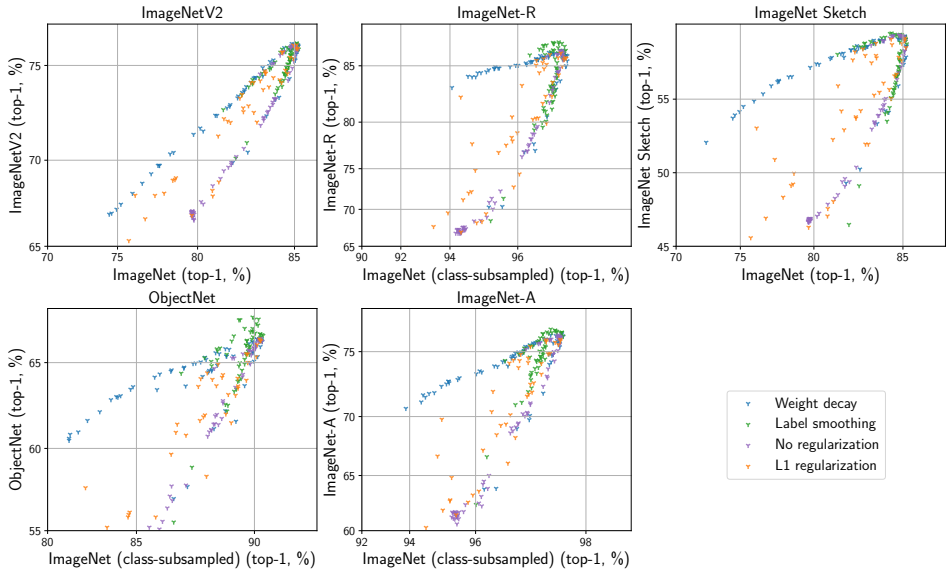

Figure 17: **Various regularizers trace similar trends when fine-tuning a linear classifier.** Comparing the relative in- and out-of-distribution performance of fine-tuning a linear classifier with the various methods of regularization discussed in Section G.3.

G.3 Fine-tuning a linear classifier

This section extends the description of linear classifier training from Appendix C.3 with details on hyperparameters and additional analyses. In each of the four regularization strategies—no regularization, weight decay, L1 regularization, and label smoothing—we run 64 hyperparameter configurations. For each trial, mini-batch size is drawn uniformly from $\{64, 128, 256\}$ and learning rate is set to $10^{-\beta}$ with β chosen uniformly at random from the range $[0, 4]$. Hyperparameters for each regularization strategy are as follows: (i) The weight decay coefficient is set to $10^{-\lambda}$ where λ is chosen uniformly at random from $[0, 4]$ for each trial; (ii) The L1 regularization coefficient is set to $10^{-\lambda}$ where λ is chosen uniformly at random from $[4, 8]$ for each trial; (iii) The label smoothing [63] coefficient λ is chosen uniformly at random from $[0, 0.25]$ for each trial. The linear classifier used for ensembling attains the best performance in-distribution. The hyperparameters from this trial are then used in the distillation and regularization experiments described in Appendix C.3. In the low-data regime (Section C.5), this process is repeated for each k and dataset.

Figure 17 demonstrates that various regularization strategies largely move along the same parabolic trend—even linear classifiers trained without explicit regularization. Figure 18 demonstrates that in the absence of explicit regularization, increasing the learning rate moves monotonically along this trend.

When training linear classifiers with k images per class as in Section C.5 the number of epochs is scaled approximately inversely proportional to the amount of data removed (e.g., with half the data we train for twice as many epochs so the number of iterations is consistent). To choose the number of epochs we use default PyTorch AdamW hyperparameters (learning rate 0.001, weight decay 0.01) and double the number of epochs until performance saturates.

G.4 ObjectNet

The zero-shot models in Table 2 use the ImageNet class names instead of the ObjectNet class names. However, this *adaptation to class shift* improves performance by 2.3% [72]. Out of the 5 datasets used for the majority of the experiments in Section **??**, ObjectNet is the only dataset for which this is possible. In Figure 19 we compare weight-space ensembles with and without adaptation to class shift.

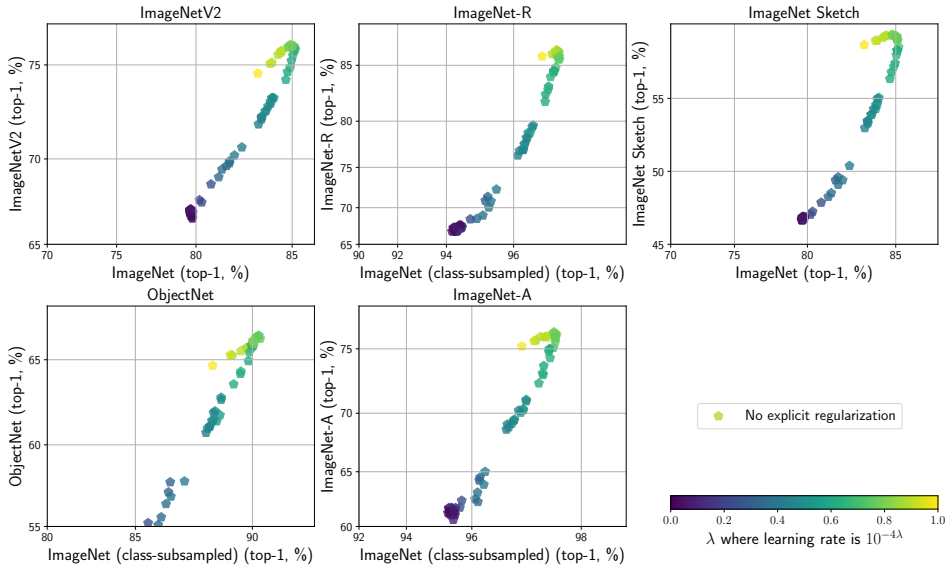

Figure 18: Comparing the relative in- and out-of-distribution performance of fine-tuning a linear classifier with various learning rates and no explicit regularization. As discussed in Section G.3, batch size is chosen randomly from $\{64, 128, 256\}$ for each experiment. As learning rate increases the linear classifiers follow a parabolic trend similar to the trend followed by explicit regularization (see Figure 17).

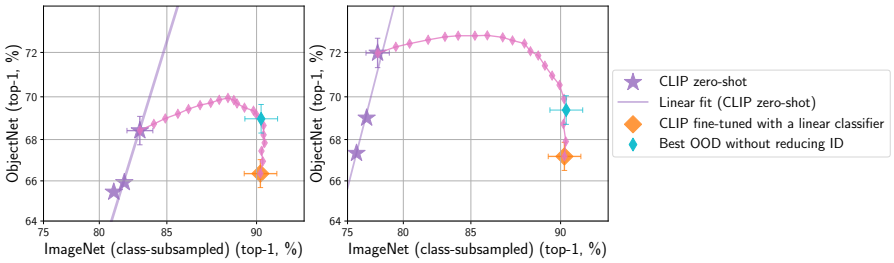

Figure 19: Effective robustness scatter plots for ObjectNet, with and without adapting to class shift. **Left:** Using ImageNet class names to construct the zero-shot classifier. **Right:** Using ObjectNet class names to construct the zero-shot classifier.

H Diversity measures

Let $\mathcal{S} = \{(x^{(i)}, y^{(i)}), 1 \leq i \leq N\}$ be a classification set with input data $x^{(i)}$ and labels $y^{(i)} \in \{1, ..., C\}$, where C is the number of classes. A classifier f is a function that maps inputs x to logits $f(x) \in \mathbb{R}^C$, yielding predictions $\hat{y} = \arg\max_{1 \leq c \leq C} f(x)_c$. We consider measures of diversity $\mathcal{M}(f, g, \mathcal{S})$ between two classifiers f and g and the dataset $\mathcal{S}$. For simplicity, $\hat{y}_f^{(i)}$ is used to denote the predictions from classifier f given inputs $x^{(i)}$ (and similarly for g).

Prediction Diversity (PD). One of the most intuitive ways to measure diversity between pairs of classifiers is to compute the fraction of samples where they disagree while one is correct [38, 81]. Formally, the prediction diversity PD is defined as:

$$\mathrm{PD}(f, g, \mathcal{S}) = \frac{1}{N} \sum_{1 \leq i \leq N} \mathbb{1}\left[\left(\hat{y}_f^{(i)} = y^{(i)} \wedge \hat{y}_g^{(i)} \neq y^{(i)}\right) \vee \left(\hat{y}_f^{(i)} \neq y^{(i)} \wedge \hat{y}_g^{(i)} = y^{(i)}\right)\right]. \tag{3}$$

Cohen's Kappa Complement (CC). Cohen's kappa coefficient is a measure of agreement between two annotators [60]. Here, we use it's complement as a diversity measure between two classifiers:

$$\mathrm{CC}(f, g, \mathcal{S}) = 1 - \frac{p_o - p_e}{1 - p_e} = \frac{1 - p_o}{1 - p_e}, \tag{4}$$

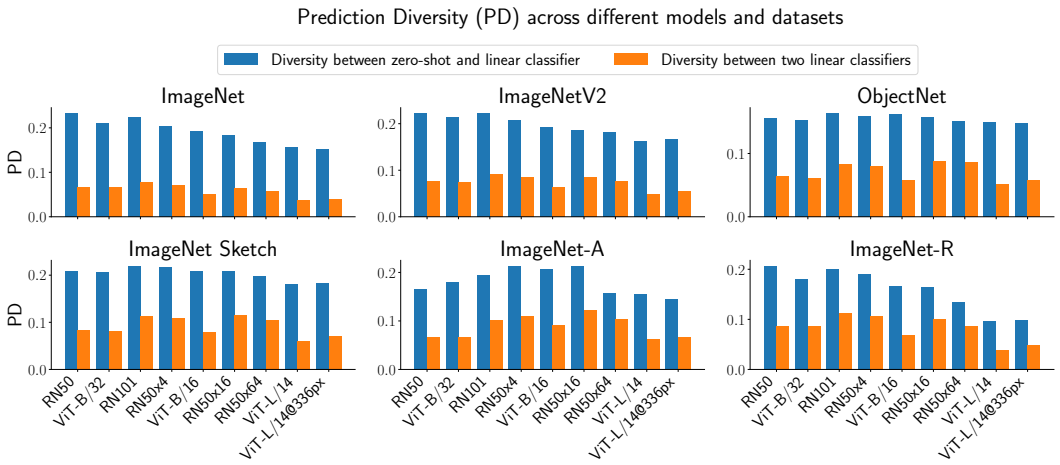

Figure 20: **Prediction Diversity (PD)** for multiple datasets and CLIP models (Equation 3).

where p_e is the expected agreement between the classifiers and p_o is the empirical probability of agreement. Formally, if $n_{f,k}$ is the number of samples where classifier f predicted label k (i.e. $n_{f,k} = \sum_{1 \leq i \leq N} \mathbb{1}[\hat{y}_f^i = k]$), then:

$$p_e = \frac{1}{N^2} \sum_{1 \leq c \leq C} n_{f,c} n_{g,c}, \quad p_o = \frac{1}{N} \sum_{1 \leq i \leq N} \mathbb{1}[\hat{y}_f^i = \hat{y}_g^i] \tag{5}$$

KL Divergence (KL). The Kullback-Leibler divergence measures how different a probability distribution is from another. Let $p_f^{(i)} = \text{softmax}\left(f(x^{(i)})\right)$ for a classifier f, and let $p_{f,c}^{(i)}$ be the probability assigned to class c. We consider the average KL-divergence over all samples as a diversity measure:

$$\text{KL}(f, g, \mathcal{S}) = \frac{1}{N} \sum_{1 \leq i \leq N} \sum_{1 \leq c \leq C} p_{f,c}^{(i)} \log \left(\frac{p_{f,c}^{(i)}}{p_{g,c}^{(i)}} \right). \tag{6}$$

Centered Kernel Alignment Complement (CKAC). CKA is a similarity measure that compares two different sets of high-dimensional representations [46]. It is commonly used for comparing representations of two neural networks, or determining correspondences between two hidden layers of the same network. CKA measures the agreement between two matrices containing the pair-wise similarities of all samples in a dataset, where each matrix is constructed according to the representations of a model. More formally, let $S \in \mathbb{R}^{N \times d}$ denote the d-dimensional features for all samples in a dataset $\mathcal{S}$, pre-processed to center the columns. For two models f and g yielding similarity matrices S_f and S_g, CKA is defined as:

$$\text{CKA}(f, g, \mathcal{S}) = \frac{||S_g^\top S_f||_F^2}{||S_f^\top S_f||_F^2 ||S_g^\top S_g||_F^2}, \tag{7}$$

where $||S||_F$ denotes the Frobenius norm of the matrix S. Larger CKA values indicate larger similarities between the representations of the two models, and thus, smaller diversity. We define the diversity measure CKAC as:

$$\text{CKAC} = 1 - \text{CKA}. \tag{8}$$

Note that CKAC is computationally expensive to compute for large datasets. For this reason, in our experiments with distributions larger than 10,000 samples, we randomly sample 10,000 to compute this measure.

Diversity across different architectures We extend Figure 15 to show results for all combinations of diversity measures, datasets, and CLIP models. Similarly to before, the baselines compares models with the same encoder, with two linear classifiers trained on different subsets of ImageNet with half of the data. Results are shown in Figures 20-23.

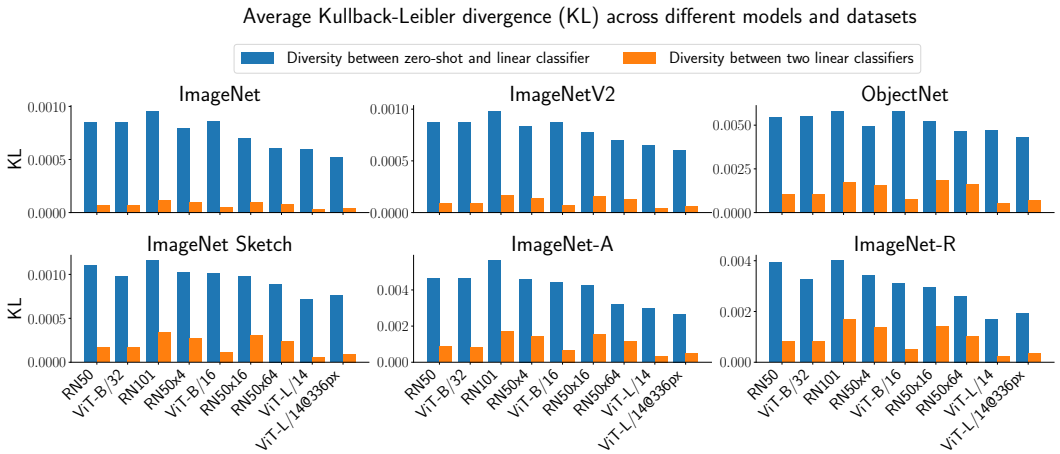

Figure 21: **Cohen's Kappa Complement (CC)** for multiple datasets and CLIP models (Equation 4).

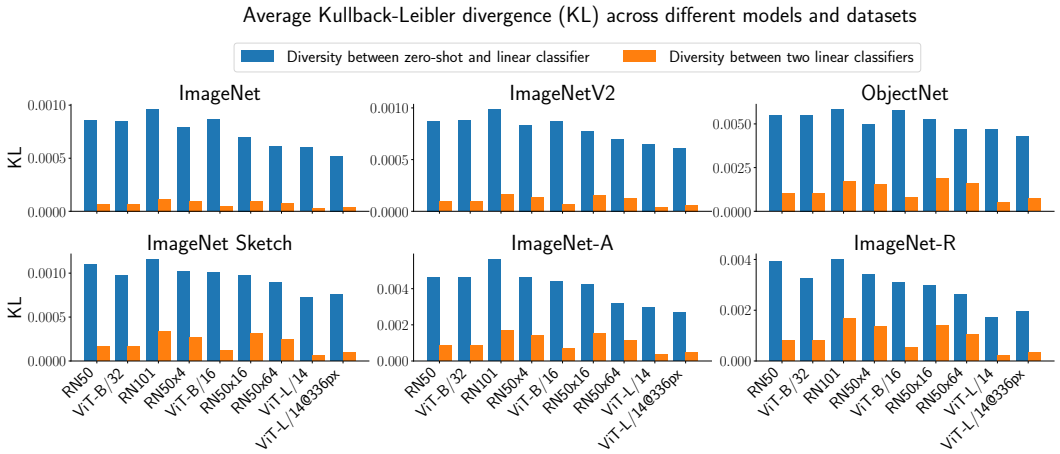

Figure 22: **Average KL Divergence (KL)** for multiple datasets and CLIP models (Equation 6).

I When do weight-space ensembles approximate output-space ensembles?

In practice we observe a difference between weight-space and output-space ensembling. However, it is worth noting that these two methods of ensembling are not as different as they initially appear. In certain regimes a weight-space ensemble approximates the corresponding output-space ensemble—for instance, when training is well approximated by a linear expansion, referred to as the NTK regime [40]. Fort et al. [24] find that a linear expansion becomes more accurate in the later phase of neural network training, a phase which closely resembles fine-tuning.

Consider the set $\Theta = \{(1-\alpha)\theta_0 + \alpha\theta_1 : \alpha \in [0,1]\}$ consisting of all θ which lie on the linear path between θ_0 and θ_1.

Proposition 1. When $f(\theta) = f(\theta_0) + \nabla f(\theta_0)^\top(\theta - \theta_0)$ for all $\theta \in \Theta$, the weight- and output-space ensemble of θ_0 and θ_1 are equivalent.

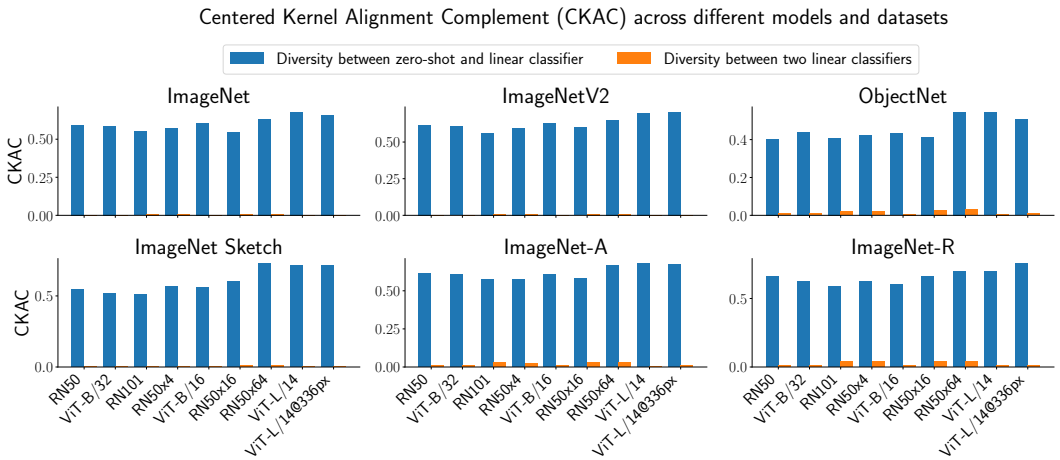

Figure 23: **Central Kernel Alignment Complement (CKAC)** for multiple datasets and CLIP models (Equation 8).

Proof. We may begin with the weight-space ensemble and retrieve the output-space ensemble

$$
\begin{aligned}
& f((1-\alpha)\theta_0 + \alpha\theta_1) && (9)\\
&= f(\theta_0) + \nabla f(\theta_0)^\top((1-\alpha)\theta_0 + \alpha\theta_1 - \theta_0) && (10)\\
&= f(\theta_0) + \alpha\nabla f(\theta_0)^\top(\theta_1 - \theta_0) && (11)\\
&= f(\theta_0) + \alpha\nabla f(\theta_0)^\top(\theta_1 - \theta_0) + \alpha f(\theta_0) - \alpha f(\theta_0) && (12)\\
&= (1-\alpha)f(\theta_0) + \alpha\Big(f(\theta_0) + \nabla f(\theta_0)^\top(\theta_1 - \theta_0)\Big) && (13)\\
&= (1-\alpha)f(\theta_0) + \alpha f(\theta_1) && (14)
\end{aligned}
$$

where the first and final line follow by the linearity assumption. □

J Additional Figures

This section provides supplemental figures:

- We compare weight-space ensembling to a series of alternatives as in Appendix C.3 and Figure 5. However, instead of displaying average in- and out-of-distribution we show the comparison separately for each dataset (Figures 24, 25, 26, 27, and 28).
- Figures 10 and 11 show a breakdown of in-distribution gains in the low-data regime for the seven datasets averaged in Figure 9.
- Figure 29 provides a schematic for the average error landscape.

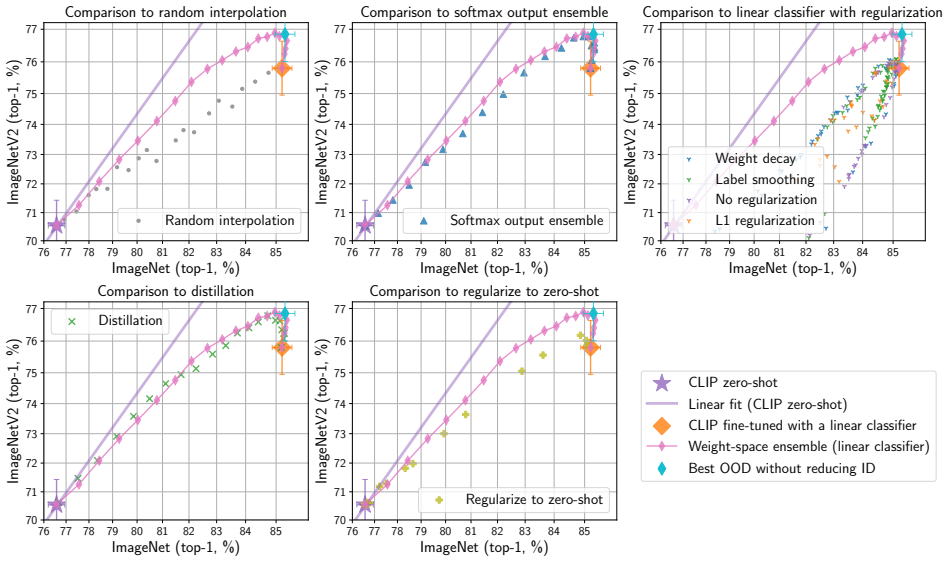

Figure 24: Comparing the relative in- and out-of-distribution performance of WiSE-FT with the alternatives described in Appendix C.3 on ImageNetV2.

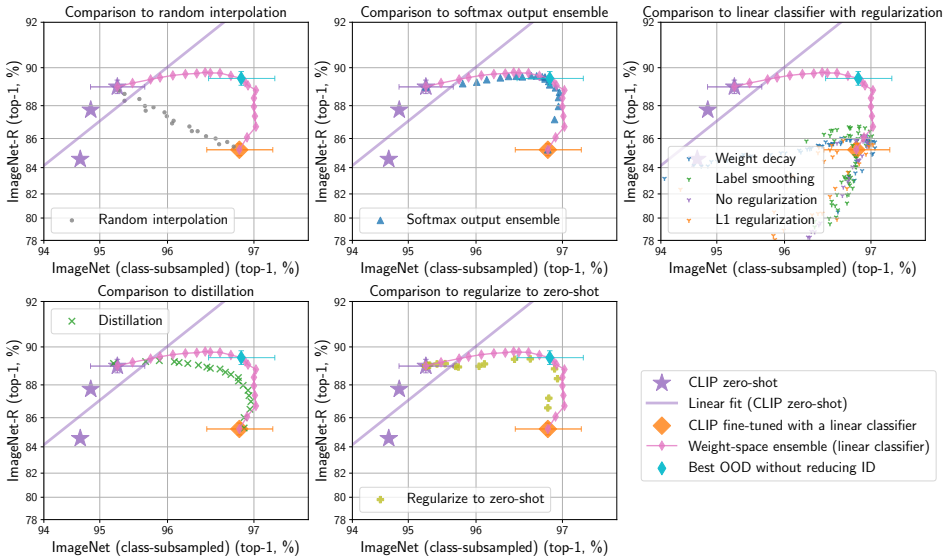

Figure 25: Comparing the relative in- and out-of-distribution performance of WiSE-FT with the alternatives described in Appendix C.3 on ImageNet-R.

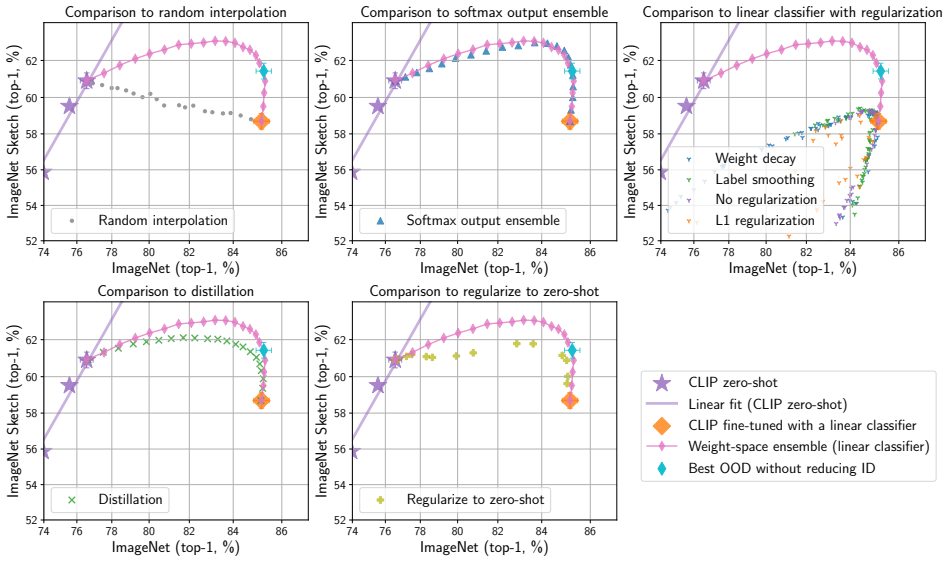

Figure 26: Comparing the relative in- and out-of-distribution performance of WiSE-FT with the alternatives described in Appendix C.3 on ImageNet Sketch.

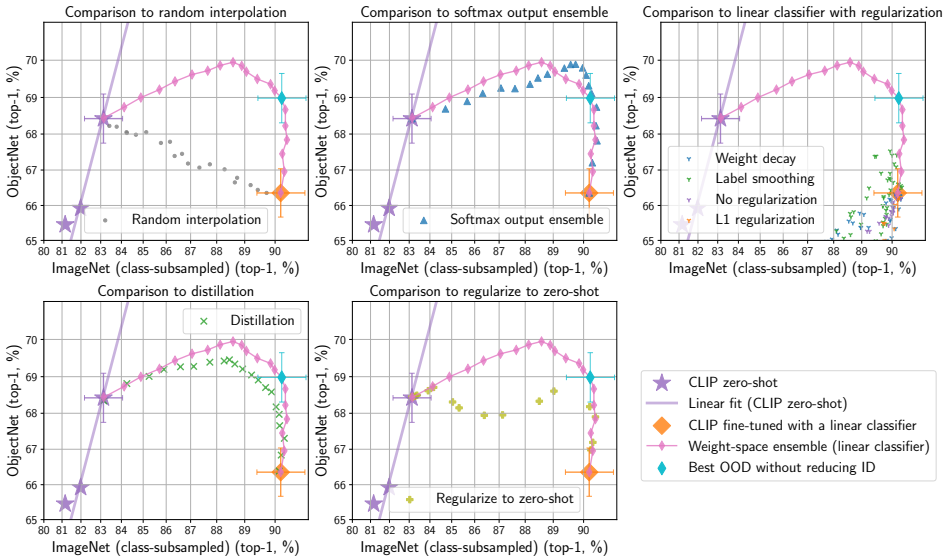

Figure 27: Comparing the relative in- and out-of-distribution performance of WiSE-FT with the alternatives described in Appendix C.3 on ObjectNet.

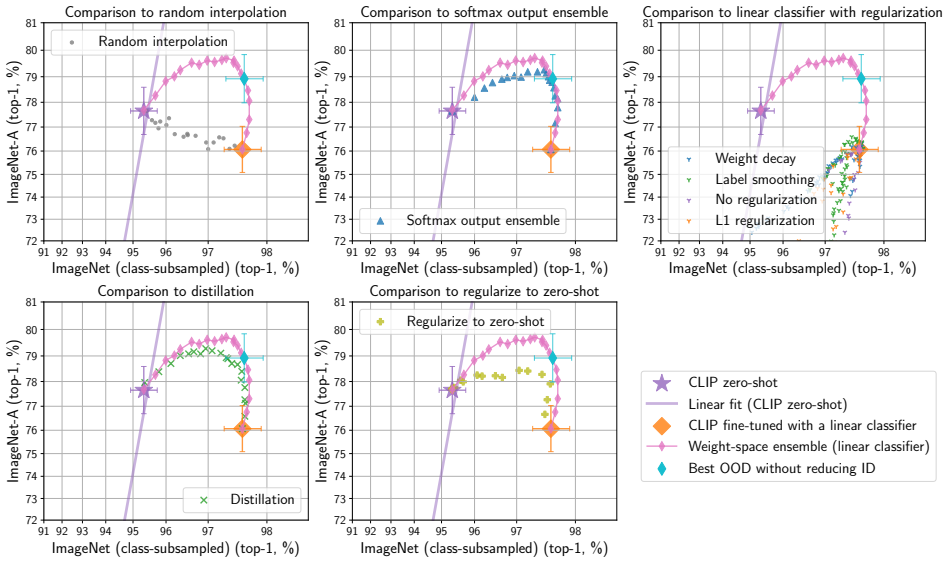

Figure 28: Comparing the relative in- and out-of-distribution performance of WiSE-FT with the alternatives described in Appendix C.3 on ImageNet-A.

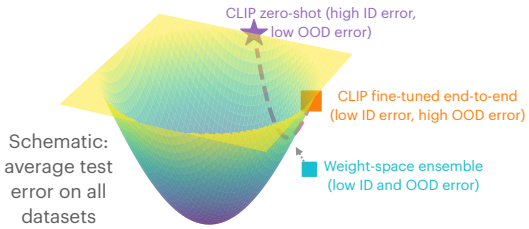

Figure 29: Schematic of the average error landscape. Li et al. [52] observe that solution spaces for a given task are high dimensional, while D'Amour et al. [16], Wortsman et al. [90] observe that movement within the solution space can change model performance on other data distributions. According to these observations, it is possible that the model finds a solution that performs well on the downstream task during fine-tuning without leaving a region of low error on the original task. Moreover, interpolating between the two solutions may travel closer to a true minimum [65, 39].