AutoFT: Robust Fine-Tuning by Optimizing Hyperparameters on OOD Data

Caroline Choi* Stanford University
Yoonho Lee* Stanford University
Annie S. Chen Stanford University
Allan Zhou Stanford University
Aditi Raghunathan Carnegie Mellon University
Chelsea Finn Stanford University

Abstract

Foundation models encode rich representations that can be adapted to a desired task by fine-tuning on task-specific data. However, fine-tuning a model on one particular data distribution often compromises the model's original performance on other distributions. Current methods for robust fine-tuning utilize hand-crafted regularization techniques to constrain the fine-tuning process towards the base foundation model. Yet, it is hard to directly specify what characteristics of the foundation model to retain during fine-tuning, as this is influenced by the complex interplay between the pre-training, fine-tuning, and evaluation distributions. We propose AutoFT, a data-driven approach for guiding foundation model fine-tuning. AutoFT optimizes fine-tuning hyperparameters to maximize performance on a small out-of-distribution (OOD) validation set. To guide fine-tuning in a granular way, AutoFT searches a highly expressive hyperparameter space that includes weight coefficients for many different losses, in addition to learning rate and weight decay values. We evaluate AutoFT on four natural distribution shifts, which include domain shifts and subpopulation shifts. Our experiments show that AutoFT significantly improves generalization to new OOD data, outperforming existing robust fine-tuning methods. Notably, AutoFT achieves a new state-of-the-art on the iWildCam benchmark, outperforming the previous best method by 4.6%.

1 Introduction

Foundation models have emerged as a powerful tool in machine learning, demonstrating unprecedented performance across a wide variety of data distributions (Radford et al., 2021a; Ilharco et al., 2021; Jia et al., 2021). By pre-training on large and diverse datasets, these models learn representations that can serve as rich common-sense priors that complement task-specific data. We thus expect fine-tuning to enhance the generalization capabilities of foundation models. However, fine-tuning often degrades the performance of foundation models on out-of-distribution (OOD) data. This indicates that conventional fine-tuning strategies can fail to utilize the prior knowledge embedded in the foundation model.

This issue of conventional fine-tuning distorting beneficial foundation model priors has driven recent research on developing *robust* fine-tuning methods. Such methods aim to produce an adapted model that achieves good performance under distribution shifts by preserving the prior knowledge embedded in the foundation model. Prior works have proposed various regularization techniques for this purpose, such as ensembling models before and after adaptation (Wortsman et al., 2022b) or initially fitting only the last layer (Kumar et al., 2022a). However, as these methods are primarily based on human intuition, they may not fully account for the complex interplay between the foundation model priors and the adaptation process.

Workshop on Distribution Shifts, 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

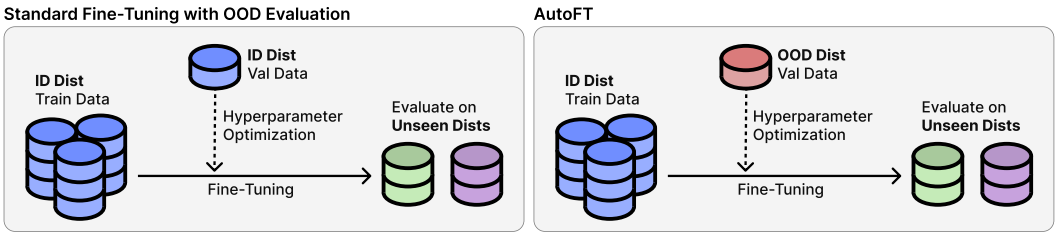

Figure 2: A summary of our data assumptions and evaluation protocol. The de facto approach is to optimize hyperparameters on a validation dataset that shares the same distribution as the training data. In contrast, AutoFT employs a small out-of-distribution (OOD) validataion set for hyperparameter optimization, enhancing the generalizability of the final model. We evaluate all fine-tuned models on data from unseen distribution shifts.

We introduce AutoFT, a novel method for robust fine-tuning that aims to find the right tradeoff between the prior and the fine-tuning data through hyperparameter optimization. Our main insight is that we can *learn* what characteristics of the foundation model to preserve during fine-tuning by using a data-driven approach. Like existing robust fine-tuning methods, we fine-tune a foundation model on task-specific data, and then evaluate the resulting model on a set of OOD distributions. However, we additionally leverage a small OOD validation set with up to 1000 labeled examples from one unseen distribution; we optimize fine-tuning hyperparameters for post-adaptation performance on this OOD validation set. Importantly, the OOD validation set is only used for hyperparameter optimization, not fine-tuning, and does not follow the same distribution as the OOD test sets. We illustrate the intuition behind our approach in Figure 1 and our data assumptions in Figure 2.

We make two key alterations to standard hyperparameter optimization, which we find to be critical for the setting of foundation model adaptation. First, as mentioned above, we optimize hyperparameters with respect to an OOD validation set rather than an ID validation set. Second, we use a broader definition of "hyperparameter": beyond the usual hyperparameters such as learning rate, we learn the fine-tuning objective itself through weight coefficients for several different loss functions and regularizers. This larger hyperparameter search space gives AutoFT more granular control over adaptation.

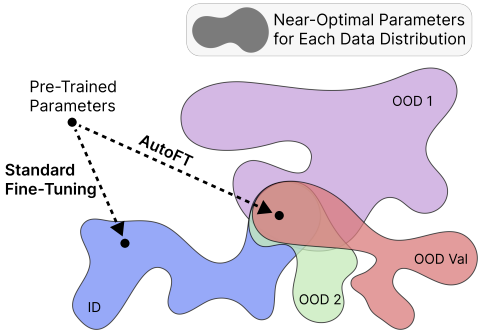

Figure 1: Overview of AutoFT. AutoFT is a data-driven approach for effectively adapting to new data during fine-tuning, while preserving pretrained model priors. AutoFT optimizes an expansive set of hyperparameters on a small validation set.

We rigorously evaluate AutoFT on a wide array of real-world datasets and consider various types of distribution shift, including subpopulation shift and domain shift. Our experiments show that our approach results in better generalization to unseen OOD data. With only 1000 (or fewer) datapoints from an OOD distribution, AutoFT outperforms existing robust fine-tuning methods across all benchmarks. These gains in robustness are achieved with minimal additional compute, requiring up to 5% more compute than standard fine-tuning in total. Among other results, AutoFT achieves new state-of-the-art performance on the challenging iWildCam benchmark (Beery et al., 2021; Koh et al., 2021), outperforming the prior best method by 4.6%.

2 Background: Hyperparameter Optimization

We begin by formalizing hyperparameter optimization, a procedure we extend for robustly fine-tuning foundation models in Section 3. Hyperparameters are predefined properties of the learning algorithm which are not learned during training, such as network architecture, learning rate, and regularization strength. Since hyperparameters significantly impact the performance of the final model, it is crucial to choose the right hyperparameters is crucial for any learning problem. The

optimal hyperparameters are hard to know beforehand as they are influenced by many properties of the problem setting, including the data distribution and desired performance metric.

Formally, we denote the the learning algorithm as LearnAlg and its hyperparameters as $\phi \in \Phi$, where Φ is the hyperparameter space. We also denote the training and validation datasets as D_{tr} and D_{val}, respectively. These datasets are disjoint, and are typically drawn from the same distribution. We denote the resulting model as $\text{LearnAlg}(\phi, D_{\text{tr}})$, to explicitly represent the dependence on both hyperparameters ϕ and training data D_{tr}. The goal of hyperparameter optimization is to find hyperparameters that maximize some performance metric $\text{Perf}(f, D_{\text{val}})$ which depends on the model f and the validation dataset D_{val}. Examples of performance metrics include top-1 accuracy, macro F1 score, and worst-region accuracy. We can formally state the hyperparameter optimization problem as

$$\phi^* = \arg\max_{\phi \in \Phi} \mathbb{E}\big[\overbrace{\text{Perf}(\underbrace{\text{LearnAlg}(\phi, D_{\text{tr}})}_{\text{Learned Parameters}}, D_{\text{val}})}^{\text{Validation Set Performance}} \big]. \tag{1}$$

Here, the expectation is taken over any randomness in the learning algorithm LearnAlg such as input data shuffling or random initialization. The optimized hyperparameters ϕ^* are subsequently used to train the final model.

Prior hyperparameter optimization methods typically start with randomly initialized model parameters and use a validation set D_{val}, drawn from the same distribution as the training data, to adjust hyperparameters. The problem of robust fine-tuning, however, begins with pre-trained model parameters and aims to achieve high performance on test data that diverges from the original training data's distribution. In the next section, we describe how we modify the standard hyperparameter optimization loop for robustly fine-tuning pre-trained foundation models.

3 AutoFT: Robust Fine-Tuning via Hyperparameter Optimization

We consider the problem of hyperparameter optimization and model selection for OOD generalization. To adapt hyperparameter optimization for robust fine-tuning, we utilize three core insights. First, we consider a larger hyperparameter space. We extend the definition of "hyperparameter" to include (per-layer) weight coefficients for nine varied loss functions and regularizers, learning rates, and weight decays. Second, we perform hyperparameter optimization on an OOD validation set rather than an ID validation set. Third, performance on one OOD distribution serves as a good proxy for performance on other unseen OOD distributions. This eliminates the need to optimize hyperparameters over an exhaustive list of OOD datasets, reducing the complexity and computational cost of hyperparameter optimization. AutoFT leverages a small held-out OOD set to guide the hyperparameter optimization process.

Data assumptions. We consider the setting of adapting a foundation model to a new task by fine-tuning on task-specific data, with the goal of achieving good performance across naturally-occuring distribution shifts in the task-specific data. During training, we assume access to datasets from related distributions: (1) a large fine-tuning dataset D_{tr} from the training distribution $\mathcal{P}_{\text{tr}}$ and (2) a small held-out OOD validation set D_{val} from a shifted distribution $\mathcal{P}_{\text{val}}$. We note that D_{val} is much smaller than D_{tr} and is only used for hyperparameter optimization, not for fine-tuning. At test time, we evaluate the model on several OOD test distributions $\mathcal{P}_{\text{ood}}$, which are different from both $\mathcal{P}_{\text{tr}}$ and $\mathcal{P}_{\text{val}}$, and are unseen during training and hyperparameter optimization.

Hyperparameter optimization for OOD generalization. Let f denote a pretrained foundation model with parameters θ, which we will adapt to the task at hand by fine-tuning on D_{tr}. Let $\phi \in \Phi$ represent fine-tuning hyperparameters, and let $\text{LearnAlg}(\phi, D_{\text{tr}})$ denote the fine-tuning algorithm which produces adapted parameters. We optimize the hyperparameters ϕ such that the fine-tuned model has good performance on the OOD validation set D_{val}:

$$\phi^* = \arg\max_{\phi \in \Phi} \mathbb{E}[\text{Perf}(\text{LearnAlg}(\phi, D_{\text{tr}}), D_{\text{val}})]. \tag{2}$$

In this work, we consider stochastic gradient descent (SGD) as the fine-tuning algorithm LearnAlg and classification accuracy as the performance measure Perf, but other algorithms and metrics can be used as well. We summarize the hyperparameter optimization procedure in Algorithm 1.

Expanded hyperparameter space. For the purposes of robust fine-tuning, we consider a more larger hyperparameter space Φ than what is typically considered in hyperparameter optimization.

Methods	iWILDCam				FMoW			
	Without Ensembling		With Ensembling		Without Ensembling		With Ensembling	
	ID	OOD	ID	OOD	ID	OOD	ID	OOD
Zeroshot	8.7 (-)	11.0 (-)	8.7 (-)	11.0 (-)	20.4 (-)	18.7 (-)	20.4 (-)	18.7 (-)
LP	44.5 (0.6)	31.1 (0.4)	45.5 (0.6)	31.7 (0.4)	48.2 (0.1)	30.5 (0.3)	48.5 (0.1)	30.7 (0.3)
FT	48.1 (0.5)	35.0 (0.5)	48.1 (0.5)	35.0 (0.5)	68.5 (0.1)	39.2 (0.7)	68.5 (0.1)	39.2 (0.7)
L2-SP	48.6 (0.4)	35.3 (0.3)	48.6 (0.4)	35.3 (0.3)	68.6 (0.1)	39.4 (0.6)	68.6 (0.1)	39.4 (0.6)
LP-FT	49.7 (0.5)	34.7 (0.4)	50.2 (0.5)	35.7 (0.4)	68.4 (0.2)	40.4 (1.0)	68.4 (0.2)	40.4 (1.0)
FLYP	**52.2 (0.6)**	35.6 (1.2)	**52.5 (0.6)**	37.1 (1.2)	**68.6 (0.2)**	41.3 (0.8)	**69.0 (0.1)**	41.9 (0.7)
AUTOFT	51.0 (0.5)	**38.3 (0.5)**	51.3 (0.5)	**39.3 (0.5)**	67.1 (0.3)	**42.3 (0.5)**	67.1 (0.3)	**42.3 (0.5)**

Table 1: AutoFT outperforms all baselines both with and without ensembling. Without ensembling, AutoFT improves OOD performance by 3.7% on WILDS-iWildCam and 6.1% on WILDS-FMoW, respectively. On WILDS-iWildCam, these improvements are preserved with ensembling, leading to an increase of 2.2% in OOD macro F1.

The hyperparameter space includes weight coefficients for nine different loss functions and regularizers: cross-entropy loss, hinge loss, entropy, confidence minimization on incorrect predictions, L1 norm, L2 norm, L1 distance to initial parameters, L2 distance to initial parameters, and a contrastive image-text CLIP loss. We denote these weight coefficients as $W = \{w_1, w_2, \ldots, w_9\}$: each w_i determines how much each corresponding loss function or regularizer contributes to the total loss. Denoting the i-th loss function or regularizer as $\mathcal{L}_i$, the total loss $\mathcal{L}$ is the weighted sum $\mathcal{L} = \sum_{i=1}^{9} w_i \mathcal{L}_i$. We also include the learning rate η and weight decay δ as hyperparameters. The complete set of hyperparameters is the tuple $\phi = (W, \eta, \delta)$.

Hyperparameter optimization. We employ the Tree-structured Parzen Estimator (TPE) for hyperparameter optimization. TPE is a Bayesian optimization method that utilizes a probabilistic model to sample the most promising hyperparameters to test, significantly reducing the computational burden compared to random search. We use the `optuna` library, which has an efficient open-source implementation of the TPE method (Akiba et al., 2019). Each hyperparameter is sampled from an appropriately scaled distribution as follows:

$$w_i \sim \text{LogUniform}(w_{\min}, w_{\max}), \quad \eta \sim \text{Uniform}(\eta_{\min}, \eta_{\max}), \quad \delta \sim \text{Uniform}(0.0, 1.0). \tag{3}$$

We find that in practice, $(w_{\min}, w_{\max}) = (1\text{e-}4, 1\text{e+}2)$ is a good range for all w_i. Given a model's conventional learning rate η^*, we set $(\eta_{\min}, \eta_{\max}) = (10^{-2}\eta^*, 10^{2}\eta^*)$.

4 Experiments

In this section, we present the main experimental findings for AutoFT. First, we show that AutoFT improves the performance of fine-tuned models on several large-scale, synthetic and natural distribution shifts, including ImageNet, WILDS-iWildCam, WILDS-FMoW, and CIFAR. Then, we present additional experiments in the low-data and transfer learning regimes. Finally, we investigate AutoFT, transferability of learned hyperparameters across fine-tuning dataset and backbone, and the effect of the choice of OOD validation distribution for hyperparameter optimization. These findings highlight the effectiveness of AutoFT in enhancing fine-tuned model performance in a variety of settings. We show detailed experimental settings in Appendix A.

4.1 Evaluation Under Distribution Shifts

Improvements on WILDS and CIFAR distribution shifts. We evaluate AutoFT on the WILDS-FMoW and WILDS-iWildCam datasets in Table 1, which present real-world distribution shits arising in satellite imagery and wildlife conservation. We additionally report results on CIFAR-10-derived distribution shifts in Table 2. Even on the more subtle CIFAR-10.1 and CIFAR-10.2 distribution shifts, we find that AutoFT outperforms fine-tuning on both metrics. AutoFT consistently outperforms all baselines on novel OOD distributions. These gains in OOD performance are maintained when ensembling zero-shot and fine-tuned models, following the method by Wortsman et al. (2022b). Weight ensembling results are with the mixture coefficient that yields the highest ID validation accuracy.

State-of-the-art performance on iWildCam. To assess whether performance gains by AutoFT continue to hold on larger foundation models, we evaluate AutoFT with a larger `ViT-L/14@336px` model. As shown in Table 3, AutoFT achieves significant gains in OOD performance of 3.6%

Method	CIFAR-10.1	CIFAR-10.2
Zero-shot	92.5	88.8
Fine-tuning	95.9	91.3
AUTOFT	**97.5**	**93.5**
WiSE-FT (best α)	98.0	94.4
AUTOFT (best α)	**98.3**	**95.0**

Table 2: AutoFT outperforms fine-tuning by 2.2% on CIFAR-10.2 and by 1.4% on CIFAR-10.1, using only 100 samples from CIFAR-10-C. AutoFT additionally outperforms WiSE-FT with weight ensembling.

	Architecture	ID	OOD
ABSGD	ResNet50	47.5 (1.6)	33.0 (0.6)
ERM	PNASNet	52.8 (1.4)	38.5 (0.6)
ERM	ViTL	55.8 (1.9)	41.4 (0.5)
Model Soups	ViTL	57.6 (1.9)	43.3 (1.0)
FLYP	ViTL-336px	**59.9 (0.7)**	46.0 (1.3)
AUTOFT	ViTL-336px	58.2 (1.0)	**50.6 (0.5)**

Table 3: AutoFT with weight ensembling attains state-of-the-art OOD performance on the WILDS-iWildCam benchmark with a `ViT-L/14-336px` backbone, surpassing the top five entries on the leaderboard (Koh et al., 2021). We observe similar performance gains by AutoFT using a smaller `ViT-B/16` architecture in Table 1.

	PatchCamelyon			SST2		
k (shots)	4	16	32	4	16	32
Zeroshot	56.5 (-)	56.5 (-)	56.5 (-)	60.5 (-)	60.5 (-)	60.5 (-)
LP	60.4 (4.0)	64.4 (3.7)	67.0 (4.4)	60.8 (1.8)	61.9 (1.4)	62.9 (1.3)
FT	63.1 (5.5)	71.6 (4.6)	75.2 (3.7)	61.1 (0.7)	62.4 (1.6)	63.4 (1.9)
LP-FT	62.7 (5.3)	69.8 (5.3)	73.9 (4.6)	60.9 (2.4)	62.9 (1.9)	63.6 (1.4)
FLYP	66.9 (5.0)	74.5 (2.0)	76.4 (2.4)	61.3 (2.7)	65.6 (2.1)	68.0 (1.7)
AUTOFT	**68.1 (5.1)**	**76.8 (2.9)**	**79.5 (2.0)**	**65.0 (3.8)**	**67.5 (1.1)**	**69.0 (1.1)**

Table 4: AutoFT shows superior performance in binary few-shot classification. AutoFT outperforms FLYP by 3.1% and full fine-tuning by 4.3% in 32-shot classification on PatchCamelyon.

over the current leader on the WILDS-iWildCam benchmark (Koh et al., 2021), FLYP (Goyal et al., 2022). AutoFT additionally outperforms the compute-intensive ModelSoups (Wortsman et al., 2022a), which ensembles more than 70 models fine-tuned with LP-FT and different augmentations and hyperparameters. AutoFT also outperforms LP-FT, a state-of-the-art baseline for fine-tuning. Even with a smaller `ViT-B/16` architecture, AutoFT outperforms all prior approaches.

4.2 Few-Shot Classification

In many real-world applications, fine-tuning often involves limited amounts of labeled, task-specific data. Few-shot classification serves as an important benchmark for evaluating the utility of fine-tuning approaches in these settings. Few-shot binary classification is a particularly challenging task for adaptation, given the small number of training examples. We evaluate on 4, 16, and 32 shot binary classification tasks from the PatchCamelyon and Rendered-SST2 datasets, following Radford et al. (2021a). PatchCamelyon contains digital pathology images for the detection of metastatic tissue. Rendered-SST2 focuses on optical character recognition for classifying text sentiment as positive or negative.

AutoFT demonstrates strong generalization capabilities with limited data, outperforming all baselines on all few-shot tasks Table 4. For example, AutoFT outperforms FLYP by 3.7% and full fine-tuning by 3.9% in a challenging 4-shot classification task on Rendered-SST2.

5 Conclusion

We introduce AutoFT, a novel data-driven approach for robust fine-tuning that optimizes 12 different hyperparameters using a small validation set distinct from the fine-tuning distribution. AutoFT only requires a small amount of data from *one* naturally occurring OOD distribution—data from a non-ID distribution is often readily available or is possible to gather at a similar cost to that of the original ID training data. Our empirical results demonstrate that AutoFT consistently outperforms existing approaches for robust fine-tuning on 4 real-world distribution shifts, suggesting that it is an effective approach for adapting foundation models. We hope that our work will inspire future research on data-driven approaches for robust fine-tuning.

References

Aghajanyan, A., Shrivastava, A., Gupta, A., Goyal, N., Zettlemoyer, L., and Gupta, S. (2020). Better fine-tuning by reducing representational collapse. *arXiv preprint arXiv:2008.03156*. [page 13]

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631. [page 4]

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and De Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29. [page 13]

Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., and Katz, B. (2019). Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32. [page 12]

Beery, S., Agarwal, A., Cole, E., and Birodkar, V. (2021). The iwildcam 2021 competition dataset. *arXiv preprint arXiv:2105.03494*. [page 2, 11, 12]

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2). [page 13]

Chen, X., Liang, C., Huang, D., Real, E., Wang, K., Liu, Y., Pham, H., Dong, X., Luong, T., Hsieh, C.-J., et al. (2023). Symbolic discovery of optimization algorithms. *arXiv preprint arXiv:2302.06675*. [page 13]

Christie, G., Fendley, N., Wilson, J., and Mukherjee, R. (2018). Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6172–6180. [page 11, 12]

Cohen, N., Gal, R., Meirom, E. A., Chechik, G., and Atzmon, Y. (2022). " this is my unicorn, fluffy": Personalizing frozen vision-language representations. *arXiv preprint arXiv:2204.01694*. [page 13]

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123. [page 13]

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. (2020). Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703. [page 13]

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee. [page 12]

Eastwood, C., Mason, I., and Williams, C. K. (2022). Unit-level surprise in neural networks. In *I (Still) Can't Believe It's Not Better! Workshop at NeurIPS 2021*, pages 33–40. PMLR. [page 13]

Eastwood, C., Mason, I., Williams, C. K., and Schölkopf, B. (2021). Source-free adaptation to measurement shift via bottom-up feature restoration. *arXiv preprint arXiv:2107.05446*. [page 13]

Evci, U., Dumoulin, V., Larochelle, H., and Mozer, M. C. (2022). Head2toe: Utilizing intermediate representations for better transfer learning. In *International Conference on Machine Learning*, pages 6009–6033. PMLR. [page 13]

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., and Hutter, F. (2015). Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28. [page 13]

Gouk, H., Hospedales, T., and massimiliano pontil (2021). Distance-based regularisation of deep networks for fine-tuning. In *International Conference on Learning Representations*. [page 13]

Goyal, S., Kumar, A., Garg, S., Kolter, Z., and Raghunathan, A. (2022). Finetune like you pretrain: Improved finetuning of zero-shot vision models. *arXiv preprint arXiv:2212.00638*. [page 5, 11, 12]

Guo, Y., Shi, H., Kumar, A., Grauman, K., Rosing, T., and Feris, R. (2019). Spottune: transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4805–4814. [page 13]

Hataya, R., Zdenek, J., Yoshizoe, K., and Nakayama, H. (2020). Faster autoaugment: Learning augmentation strategies using backpropagation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 1–16. Springer. [page 13]

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al. (2021a). The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349. [page 12]

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. (2021b). Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271. [page 12]

Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5*, pages 507–523. Springer. [page 13]

Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., and Schmidt, L. (2021). Openclip. If you use this software, please cite it as below. [page 1, 11]

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. (2021). Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR. [page 1]

Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Zhao, T. (2019). Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *arXiv preprint arXiv:1911.03437*. [page 13]

Karani, N., Erdil, E., Chaitanya, K., and Konukoglu, E. (2021). Test-time adaptable neural networks for robust medical image segmentation. *Medical Image Analysis*, 68:101907. [page 13]

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526. [page 13]

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. (2021). Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR. [page 2, 5, 11, 12]

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images. [page 11]

Kumar, A., Raghunathan, A., Jones, R. M., Ma, T., and Liang, P. (2022a). Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*. [page 1, 11, 12, 13]

Kumar, A., Shen, R., Bubeck, S., and Gunasekar, S. (2022b). How to fine-tune vision models with sgd. [page 13]

Lee, C., Cho, K., and Kang, W. (2019a). Mixout: Effective regularization to finetune large-scale pretrained language models. *arXiv preprint arXiv:1909.11299*. [page 13]

Lee, J., Tang, R., and Lin, J. (2019b). What would elsa do? freezing layers during transformer fine-tuning. *arXiv preprint arXiv:1911.03090*. [page 13]

Lee, Y., Chen, A. S., Tajwar, F., Kumar, A., Yao, H., Liang, P., and Finn, C. (2022). Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466*. [page 13]

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. (2018a). Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32. [page 13]

Li, H., Chaudhari, P., Yang, H., Lam, M., Ravichandran, A., Bhotika, R., and Soatto, S. (2020). Rethinking the hyperparameters for fine-tuning. In *International Conference on Learning Representations*. [page 13]

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *The journal of machine learning research*, 18(1):6765–6816. [page 13]

Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Hardt, M., Recht, B., and Talwalkar, A. (2018b). Massively parallel hyperparameter tuning. [page 11, 13]

Lim, S., Kim, I., Kim, T., Kim, C., and Kim, S. (2019). Fast autoaugment. *Advances in Neural Information Processing Systems*, 32. [page 13]

Liu, H., Simonyan, K., and Yang, Y. (2018). Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*. [page 13]

Liu, Y., Agarwal, S., and Venkataraman, S. (2021). Autofreeze: Automatically freezing model blocks to accelerate fine-tuning. *arXiv preprint arXiv:2102.01386*. [page 13]

Lu, S., Nott, B., Olson, A., Todeschini, A., Vahabi, H., Carmon, Y., and Schmidt, L. (2020). Harder or different? a closer look at distribution shift in dataset reproduction. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, volume 5, page 15. [page 11]

Metz, L., Harrison, J., Freeman, C. D., Merchant, A., Beyer, L., Bradbury, J., Agrawal, N., Poole, B., Mordatch, I., Roberts, A., et al. (2022). Velo: Training versatile learned optimizers by scaling up. *arXiv preprint arXiv:2211.09760*. [page 13]

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724. [page 13]

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021a). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR. [page 1, 5, 11, 12]

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021b). Learning transferable visual models from natural language supervision. [page 12]

Ramasesh, V. V., Dyer, E., and Raghu, M. (2020). Anatomy of catastrophic forgetting: Hidden representations and task semantics. *arXiv preprint arXiv:2007.07400*. [page 13]

Real, E., Aggarwal, A., Huang, Y., and Le, Q. V. (2019). Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789. [page 13]

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (2018). Do cifar-10 classifiers generalize to cifar-10? [page 11]

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (2019). Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR. [page 12]

Royer, A. and Lampert, C. (2020). A flexible selection scheme for minimum-effort transfer learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2191–2200. [page 13]

Sagawa, S., Koh, P. W., Lee, T., Gao, I., Xie, S. M., Shen, K., Kumar, A., Hu, W., Yasunaga, M., Marklund, H., Beery, S., David, E., Stavness, I., Guo, W., Leskovec, J., Saenko, K., Hashimoto, T., Levine, S., Finn, C., and Liang, P. (2022). Extending the WILDS benchmark for unsupervised adaptation. In *International Conference on Learning Representations*. [page 11, 12]

Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813. [page 13]

Shen, Z., Liu, Z., Qin, J., Savvides, M., and Cheng, K.-T. (2021). Partial is better than all: Revisiting fine-tuning strategy for few-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9594–9602. [page 13]

Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., and Schmidt, L. (2020). Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599. [page 11]

Torralba, A., Fergus, R., and Freeman, W. T. (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970. [page 11]

Touvron, H., Cord, M., El-Nouby, A., Verbeek, J., and Jégou, H. (2022). Three things everyone should know about vision transformers. *arXiv preprint arXiv:2203.09795*. [page 13]

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. (2019). Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518. [page 12]

Wichrowska, O., Maheswaranathan, N., Hoffman, M. W., Colmenarejo, S. G., Denil, M., de Freitas, N., and Sohl-Dickstein, J. (2017). Learned optimizers that scale and generalize. [page 13]

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. (2022a). Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR. [page 5, 11, 13]

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Lopes, R. G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al. (2022b). Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971. [page 1, 4, 11, 12, 13]

Xu, Y., Xie, L., Zhang, X., Chen, X., Qi, G.-J., Tian, Q., and Xiong, H. (2019). Pc-darts: Partial channel connections for memory-efficient architecture search. *arXiv preprint arXiv:1907.05737*. [page 13]

Xuhong, L., Grandvalet, Y., and Davoine, F. (2018). Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*, pages 2825–2834. PMLR. [page 13]

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27. [page 13]

Zhang, J. O., Sax, A., Zamir, A., Guibas, L., and Malik, J. (2020). Side-tuning: a baseline for network adaptation via additive side networks. In *European Conference on Computer Vision*, pages 698–714. Springer. [page 13]

Zhang, M., Marklund, H., Dhawan, N., Gupta, A., Levine, S., and Finn, C. (2021). Adaptive risk minimization: Learning to adapt to domain shift. *Advances in Neural Information Processing Systems*, 34:23664–23678. [page 13]

Zoph, B. and Le, Q. V. (2016). Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*. [page 13]

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710. [page 13]

A Experimental Setup

Distribution shifts. In our analysis, we focus on *natural distribution shifts*, defined by (Taori et al., 2020) as shifts arising from real-world variations, such as changes in lighting, geography, and image styles. This approach, in line with prior works on robust fine-tuning (Radford et al., 2021a; Wortsman et al., 2022b; Kumar et al., 2022a; Goyal et al., 2022), emphasizes shifts that are representative of real-world scenarios.

Our main results include four distribution shifts, including real-world distribution shifts arising in wildlife recognition and satellite imagery, and two CIFAR-10-derived distribution shifts. We evaluate on The WILDS-iWildCam dataset (Beery et al., 2021; Koh et al., 2021; Sagawa et al., 2022) presents distribution shifts arising from variations in camera trap locations, lighting conditions, and animal behaviors across various geographic regions. The WILDS-FMoW dataset (Christie et al., 2018; Koh et al., 2021; Sagawa et al., 2022) presents distribution shifts arising from changes in time, geographic locations, and land use. Finally, we evaluate on two CIFAR-10 derived distribution shifts: CIFAR-10 to CIFAR-10.1 and CIFAR-10.2, which involve subtle changes in image characteristics and composition.

Foundation models and CLIP. We fine-tune pre-trained CLIP (Radford et al., 2021a) models, including those provided in the `open-clip` repository (Ilharco et al., 2021). We use the CLIP `ViT-B/16` model from OpenAI as our default model, unless specified otherwise. Evaluation on CIFAR-10 uses the CLIP `ViT-L/14` model from OpenAI, in line with (Wortsman et al., 2022b). Our SoTA results on WILDS-iWildCam use the CLIP `ViT-L/14-336px` model from OpenAI. We use text templates used in prior work (Radford et al., 2021a; Wortsman et al., 2022b) to generate zero-shot final layer weights for all datasets.

Effective robustness and weight ensembling curves. We use the *effective robustness* framework by Taori et al. (2020) to evaluate model robustness based on accuracy exceeding a baseline trained only on the reference distribution. Linearly interpolating the weights of a fine-tuned model and a pretrained model (WiSE-FT) has been shown to improve both ID and OOD performance (Wortsman et al., 2022a). Hence, we include weight ensembling as an additional point of comparison, and interpolate the weights of models fine-tuned by each method with 10 mixing coefficients α.

Baselines. We compare AutoFT against several methods for adapting pretrained models. We include two standard transfer learning methods that minimize cross-entropy loss: linear probing (LP) and full fine-tuning (FT). We also compare with recent works in robust fine-tuning: L2-SP (Li et al., 2018b), which fine-tunes with an L2 regularization term towards pretrained weights; LP-FT (Kumar et al., 2022a), which performs linear probing followed by full fine-tuning; and FLYP (Goyal et al., 2022), which fine-tunes with the CLIP pretraining loss – a contrastive loss between image embeddings and class-descriptive prompt embeddings. We additionally evaluate all methods with weight ensembling (WiSE-FT) (Wortsman et al., 2022b), which is shown to improve OOD performance in an orthogonal way to other robust fine-tuning methods.

Training protocol. We closely follow the training details of Goyal et al. (2022) and Wortsman et al. (2022b). All methods fine-tune models with an AdamW optimizer, cosine learning rate scheduler, and a batch size of 512 for ImageNet and 256 for all other datasets. All baseline hyperparameters, such as learning rate, weight decay, and warmup length, are tuned through grid search. All methods, including AutoFT, perform early stopping based on in-distribution (ID) validation accuracy. We provide a comprehensive breakdown of the hyperparameter sweeps in the supplementary material. We emphasize that none of these methods, including AutoFT, observes any of the test OOD distributions during training. Finally, we report average metrics over 5 runs with 95% confidence intervals.

A.1 Datasets

Below, we summarize the datasets we use for evaluation, including the fine-tuning dataset (ID), the validation dataset for hyperparameter optimization, and the test OOD datasets.

- **CIFAR-10** (Krizhevsky et al., 2009) contains 60,000 images across 10 classes. We use CIFAR-10 for fine-tuning, 100 examples from CIFAR-10-C for validation, and the CIFAR-10.1 (Recht et al., 2018; Torralba et al., 2008) and CIFAR-10.2 (Lu et al., 2020) as OOD test sets.

- **ImageNet** (Deng et al., 2009) contains over a million images in 1000 categories. We use ImageNet as our ID distribution, 15000 examples from ImageNet-C for validation, and five ImageNet variations for the OOD datasets following prior works (Radford et al., 2021b; Wortsman et al., 2022b; Kumar et al., 2022a; Goyal et al., 2022): ImageNet-V2 (Recht et al., 2019), ImageNet-R (Hendrycks et al., 2021a), ImageNet-A (Hendrycks et al., 2021b), ImageNet-Sketch (Wang et al., 2019), and ObjectNet (Barbu et al., 2019).
- **WILDS-iWildCam** (Beery et al., 2021; Koh et al., 2021; Sagawa et al., 2022) is an animal image classification dataset with 182 classes. Differences in camera specifications and attributes such as background and lighting distinguish ID and OOD. We use the ID train set from Koh et al. (2021) as the fine-tuning dataset, the OOD validation set for hyperparameter optimization, and the OOD test set for evaluation.
- **WILDS-FMoW** (Christie et al., 2018; Koh et al., 2021; Sagawa et al., 2022) contains remote sensing imagery from satellites. Each image is to be classified into one among 62 categories, including labels like "impoverished settlement" and "hospital". The ID and OOD datasets differ in time of acquisition and geographic location. We use the ID train set for fine-tuning, the OOD validation set for hyperparameter optimization, and the OOD test set for evaluation.

  In all of the transfer learning datasets described below, we use a subset of the ID validation set for hyperparameter optimization. In other words, we do not use an external "OOD" set for hyperparameter optimization.
- **Caltech101** (**?**) contains images of objects from 101 different categories, including "dragonfly," "grand piano," and "saxophone."
- **StanfordCars** (**?**) features a collection of car images categorized by model, make, and year, where the task is to classify them into one of 196 types, such as "Ford Mustang Convertible 1967" or "Toyota Prius Hatchback 2009."
- **Flowers102** (**?**) consists of flower images from the UK, with the objective of classifying each image into one of 102 species, such as "oxeye daisy" or "hibiscus."
- **PatchCamelyon** (**?**) provides digital pathology images for binary classification, with the goal of identifying metastatic tumor tissues.
- **Rendered SST2** (Radford et al., 2021a) is a dataset for optical character recognition, where the task is to classify text sentiment as "positive" or "negative."

A.2 Training Details

Baselines. We closely follow the training details in Goyal et al. (2022). For all datasets excluding ImageNet, we conduct a hyper-parameter sweep across five learning rates (1e-2 to 1e-6) and five weight decay rates (0.0 to 0.4), using a batch size of 256. On ImageNet, we perform a hyperparameter sweep over three learning rates (1e-4, 1e-5, 1e-6) and two weight decay rates (0, 0.1) with a larger batch size of 512. Additionally, L2-SP demands tuning a separate regularization term λ ranging from 1e-1 to 1e-4.

We select the baseline hyper-parameters based on the highest ID validation performance. For datasets without a standard validation set, we split the training data into an 80:20 ratio to create one.

In the few-shot setting with varying k values (4, 16, 32), we report results averaged over 50 runs, each with k training and validation examples randomly drawn from the full datasets. This compensates for the higher variance due to smaller data sizes.

AutoFT. As described in Section 3, AutoFT learns weights for nine different losses on a log-uniform range $[10^{-4}, 10]$. AutoFT additionally searches for learning rate in the log-uniform range $[10^{-2} \cdot \eta^*, 10^2 \cdot \eta^*]$, where η^* is the conventional learning rate used in prior works on fine-tuning (Wortsman et al., 2022b; Kumar et al., 2022a; Goyal et al., 2022), and weight decay values in the log-uniform range $[0.0, 1.0]$.

We report the number of inner-loop gradient steps and `Optuna` trials) we use in hyperparameter optimization as follows. On WILDS-iWildCam and WILDS-FMoW, we run AutoFT with 10 inner steps, 500 `Optuna` trials, and 1000 validation examples. On CIFAR-10, we run AutoFT with

Algorithm 1 Hyperparameter Optimization Loop

Input Hyperparameter optimizer HPO
Input ID training data D_{tr}, OOD validation data D_{val}
for $\phi \leftarrow \text{HPO.Sample}()$ **do**
  $f_{ft} \leftarrow \text{LearnAlg}(D_{\text{tr}}, \phi)$ // Fine-tune model
  $p \leftarrow \text{Perf}(f_{ft}, D_{\text{val}})$ // Evaluate performance
  $\text{HPO.Update}(\phi, p)$ // Pass performance to HPO
end for
$\phi^* \leftarrow \text{HPO.Best}()$ // Get best hyperparameters
$\theta^* \leftarrow \text{LearnAlg}(D_{\text{tr}}, \phi^*)$ // Fine-tune final model

10 inner steps, 100 `Optuna` trials, and 100 validation examples. On Flowers102 and StanfordCars, we run AutoFT with 50 inner steps, 500 `Optuna` trials, and use 500 and 1000 validation examples for hyperparameter optimization, respectively. These meta-hyperparameters are selected based on performance on a held-out ID validation set. We emphasize that in the transfer learning experiments (e.g., Flowers102 and StanfordCars), we use ID validation sets for hyperparameter evluation, and do not use a separate OOD validation set. For the iWildCam SoTA results, we fine-tune the `ViT-L/14@336px` model with loss weights learned on the smaller `ViT-B/16` backbone with AutoFT. Thus these results may underestimate AutoFT's potential. There is potential for even better performance if AutoFT is applied directly to the `ViT-L/14@336px` backbone.

For the k few-shot classification setting, where $k \in \{4, 16, 32\}$, we use a k-shot validation set for hyperparameter optimization. On all k-shot SST2 and PatchCamelyon experiments, we run with 10 inner-loop gradient steps and 50 `Optuna` trials. Due to the noise , in the few-shot binary classification setting, we select the best hyperparameters from 5 AutoFT runs based on validation performance.

B Related Work

Transfer learning. Transfer learning is an effective way to obtain performant task-specific models given limited data. Early works found that features learned from pre-training on a large dataset can serve as good initial parameters for new tasks (Oquab et al., 2014; Yosinski et al., 2014; Sharif Razavian et al., 2014). Within this paradigm, many works have proposed regularization techniques for fine-tuning (Zhang et al., 2020; Xuhong et al., 2018; Lee et al., 2019a; Jiang et al., 2019; Li et al., 2020; Aghajanyan et al., 2020; Gouk et al., 2021; Shen et al., 2021; Karani et al., 2021) or different ways of selectively freezing some pre-trained parameters (Kirkpatrick et al., 2017; Lee et al., 2019b; Guo et al., 2019; Ramasesh et al., 2020; Liu et al., 2021; Royer and Lampert, 2020; Eastwood et al., 2021; Evci et al., 2022; Eastwood et al., 2022; Cohen et al., 2022; Touvron et al., 2022; Lee et al., 2022; Kumar et al., 2022b). Specifically motivated by the fact that foundation models can be *more* robust than naively fine-tuned models, recent works have focused on improving OOD performance after fine-tuning (Wortsman et al., 2022b; Kumar et al., 2022a; Wortsman et al., 2022a). We consider the same problem setting, but instead of using a hand-designed regularization method, we *learn* the fine-tuning procedure in a data-driven way, based on performance on a small OOD validation set.

AutoML and hyperparameter optimization. Our work leverages high-level ideas from the broader literature on meta-learning and hyperparameter optimization. Such methods have proposed to optimize different parts of the training pipeline, including general hyperparameters (Hutter et al., 2011; Bergstra and Bengio, 2012; Feurer et al., 2015; Li et al., 2017, 2018b), network architectures (Zoph and Le, 2016; Zoph et al., 2018; Liu et al., 2018; Real et al., 2019; Xu et al., 2019), augmentation policies (Cubuk et al., 2019; Lim et al., 2019; Hataya et al., 2020; Cubuk et al., 2020), and optimizers (Andrychowicz et al., 2016; Wichrowska et al., 2017; Metz et al., 2022; Chen et al., 2023). However, most of these works optimize for generalization within the training distribution, and do not consider robustness to distribution shifts. Existing works that optimize a training procedure for OOD generalization consider a structured few-shot adaptation setting (Li et al., 2018a; Zhang et al., 2021), limiting their scalability to large datasets. Our work learns how to best adapt a foundation model to a new task by optimizing hyperparameters on an OOD validation set.