An Empirical Study of Pre-trained Models on Out-of-distribution Generalization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Generalizing to out-of-distribution (OOD) data – that is, data from domains unseen during training – is a key challenge in modern machine learning, which has only recently received much attention. Some existing approaches propose leveraging larger models and pre-training on larger datasets. In this paper, we provide new insights in applying these approaches. Concretely, we show that larger models and larger datasets need to be *simultaneously* leveraged to improve OOD performance. Moreover, we show that using smaller learning rates during fine-tuning is critical to achieving good results, contrary to popular intuition that larger learning rates generalize better when training from scratch. We show that strategies that improve in-distribution accuracy may, counter-intuitively, lead to poor OOD performance despite strong in-distribution performance. Our insights culminate to a method that achieves state-of-the-art results on a number of OOD generalization benchmark tasks, often by a significant margin.

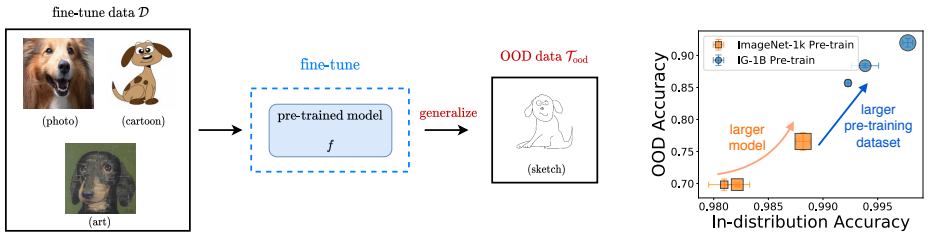
1 Introduction

Most machine learning (ML) models assume that test data is drawn from the same distribution as training data. However, this assumption does not hold in many real-world applications. As a result, ML models often fail to generalize to out-of-distribution (OOD) data encountered during their deployment and suffer from significant performance drops compared with the model performance on in-distribution (ID) data [38, 47]. For example, common distribution shifts prevalent during test time include variation in locations [22] and weather [51], noise and blur corruptions [18], and small adversarial perturbations [44]. As ML models are increasingly deployed in safety-critical applications, it is becoming ever more critical to ensure strong *OOD generalization* for such models, i.e., the models robustly generalizing to relevant OOD data not seen during training.

While this problem is indeed difficult since the goal is to generalize to data that are not seen during training, there have been a handful of methods recently proposed to improve OOD generalization. Some methods propose specialized training methods, such as simulating OOD data during training [26], learning invariant representations [2], and performing adversarial data augmentation [52]. Intriguingly, Gulrajani and Lopez-Paz [14] conducted an extensive empirical evaluation on domain generalization benchmark datasets, and demonstrated that classical empirical risk minimization (ERM) approach achieves nearly state-of-the-art OOD generalization performance compared with these specialized methods. On the other hand, most of the approaches apply small or medium size networks that are usually pre-trained on the ImageNet-1k dataset, such as pre-trained ResNet50 [14]. On the other hand, recent works find that pre-training on larger and more diverse data is one of the most effective paths toward generalizing to out-of-distribution data on ImageNet [46].

In this paper, we systematically investigate the importance of pre-trained models for OOD generalization. Specifically, we conduct extensive experiments on models with different model sizes

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



(a) Fine-tuning procedure for OOD generalization. (b) ID v.s. OOD.

Figure 1: (**Left**) Illustration of our fine-tuning procedure used throughout this paper, i.e., we fine-tune a pre-trained model on training dataset $\mathcal{D}$ and evaluate the model performance on OOD data $\mathcal{T}_{\text{ood}}$ – a data domain not seen during training. (**Right**) Evaluating ID and OOD accuracies for two classes of pre-trained models. Orange squares represent the ImageNet-1k pre-trained models and blue circles represent the IG-1B pre-trained models, where IG-1B is a larger dataset than ImageNet-1k. Larger marker size means the model size is larger.

that are pre-trained on large datasets. The pre-trained models are then fine-tuned on the training data for the underlying task. Instead of focusing on achieving state-of-the-art results on benchmark OOD datasets, our empirical study aims to develop a better understanding of the critical role that pre-trained models along with different design choices for fine-tuning such models play in ensuring good OOD generalization. This provides novel insights towards closing the gap between ID and OOD generalization for future research.

The main contributions of our work are as follows:

- When leveraging larger models for OOD generalization, we find that both large pre-training dataset sizes and large model sizes are critical. Missing either one of the two components may hurt OOD generalization.
- We show that using a small learning rate generalizes better for OOD when we leverage a pre-trained model. This is a complementary argument to Li et al. [28] that suggested to use a large learning rate for better generalization in the case of no pre-training.
- We show cases where improving in-distribution performance actually leads to worse OOD performance suggesting that ID performance is not a reliable indicator of OOD performance.
- Our insights culminate in a method that achieves SOTA, often by a significant margin, on a number of OOD generalization benchmark tasks including PACS, VLCS, and Office-Home.

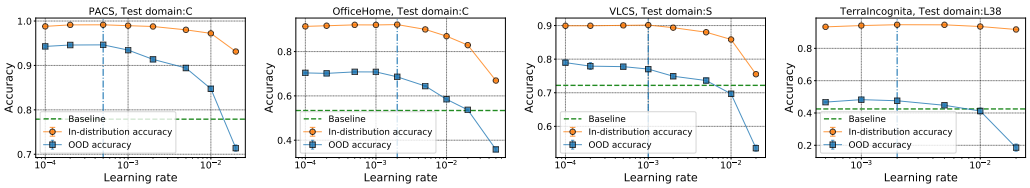
2 Main results

We present our main experimental results in this section. First, we highlight the importance of models pre-trained on large and diverse datasets for OOD generalization in Section 2.1. Next, we investigate the effect of fine-tuning learning rates, especially for models pre-trained on diverse datasets in Section 2.2. Then, we perform a systematic examination of several components of pre-trained models for OOD generalization, including model size and pre-training dataset in Section 2.3, and model architecture in Section B.1. Finally, we evaluate whether techniques used for improving ID accuracy can also enhance OOD generalization in Section B.2. Please refer to Section A for preliminaries and experimental setup.

2.1 Importance of a better pre-trained model

Models pre-trained on more diverse datasets have been shown to achieve better OOD generalization on real-world distribution shifts [46, 20]. As a warm-up for understanding the properties of pre-trained models on OOD data, we first study the OOD generalization performance of a ResNet-based model that is pre-trained on a large and diverse pre-training dataset. In particular, we focus on an SWSL-ResNext101-32x4d model pre-trained on the IG-1B-Targeted data [57], which is a much larger and more diverse dataset than ImageNet.

The fine-tuning approach described in Section A with different learning rate significantly outperforms the baseline results from Gulrajani and Lopez-Paz [14] (cf. Table 1). This shows that a better pre-trained model indeed improves OOD generalization without using specialized algorithms for domain generalization. For example, the OOD accuracy improves from 79.1% to 91.3% on **PACS** (S), and 62.7% to 76.4% on **Office-Home** (A). Overall, Table 1 suggests that using a larger model pre-trained on more data can be very effective for better OOD generalization. Next, we conduct more detailed experiments to better understand the OOD generalization performance of various pre-trained models.



(a) **PACS** (C). (b) **OfficeHome** (C). (c) **VLCS** (S). (d) **Terra** (L38).

Figure 2: Evaluating models (SWSL-ResNext101-32x4d) fine-tuned using different learning rates on ID and OOD data. (Top row) Scatter plot of ID accuracy (X-axis) and OOD accuracy (Y-axis). (Bottom row) Compare ID accuracy with OOD accuracy w.r.t. learning rate (X-axis). The green dashed line corresponds to the baseline OOD accuracy, and the blue dash-dotted line represents the selected model (by selecting the model with best ID accuracy).

2.2 Effect of fine-tune learning rate

Given the fact that simply fine-tuning the SWSL model leads to compelling improvement in the OOD generalization, we now take a closer look at models trained with different learning rates. We evaluate models trained with different learning rates on both ID and OOD test data. Figure 2 summarizes the results for fine-tuning SWSL-ResNext101-32x4d with different learning rates. In Figure 7(a)-7(d), each point in the plot corresponds to a model trained with a distinct learning rate. As mentioned in Section A, we select models that achieves the top-5 ID accuracy, and depict the standard deviation of both $\texttt{Acc}_{\text{id}}$ and $\texttt{Acc}_{\text{ood}}$ for each model. We only show results for models that achieve $> 95\%$ *training* accuracy to better compare model performance.

Regularization effect of small learning rate. Based on Figure 2, our main finding is that the fine-tuning learning rate plays a key role in determining both ID and OOD accuracy. In particular, we observe that the models trained with smaller learning rates achieve much better OOD generalization, even when the ID accuracy does not change much. This is different from the fact that larger learning rates generalize better when the model is directly trained on the training data $\mathcal{D}$ without a pre-training phase [28, 24, 35]. We also observe similar behavior with SWSL-ResNext101-32x8d models, where smaller learning rates lead to better OOD performance (see Figure 8 in Appendix). Intriguingly, we do not observe this phenomenon for models pre-trained on ImageNet. There, as shown in Figure 11 in Appendix, when the ID accuracy is similar, a model trained with larger learning rate achieves better OOD generalization (e.g., see Figure 11(e) and 11(f)). To summarize, our results indicates that when models are pre-trained on more diverse datasets, smaller learning rates can lead to better OOD generalization.

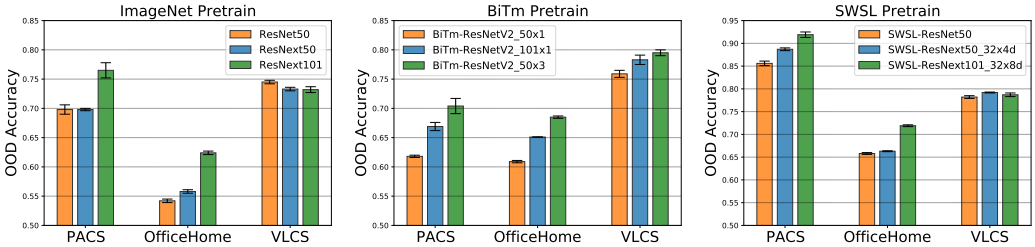
To obtain a better understanding of the effect of learning rate, we also study the OOD accuracy during training. In Figure 13, we visualize the OOD accuracy vs. training loss as measured every 100 SGD iterations for models trained with three different learning rates $\eta \in \{0.01, 0.005, 0.001\}$. Note that all three learning rates eventually achieve similar training loss, but the models trained with smaller learning rates have better OOD generalization. Figure 13 also suggests that models trained with larger learning rates cannot achieve similar OOD accuracy by using early stopping. Meanwhile, the figure also confirms the regularization effect of small learning rates for fine-tuning pre-trained models.

Overall, we find that the learning rate is a key parameter for achieving good OOD generalization. While different learning rates may not affect ID accuracy much, OOD accuracy can be very sensitive to the choice of learning rate. Our results also highlight a limitation of performing model selection based on ID accuracy as models with similar ID accuracy may have very different OOD performance.

2.3 Pre-training for better OOD generalization

Now, we systematically explore the role that pre-trained models play in improving OOD generalization. Specifically, we study three aspects of pre-trained models: pre-training dataset, model architecture, and model size. Furthermore, we compare the models trained from scratch (i.e., without pre-training) with pre-trained models with respect to OOD generalization.

Effect of pre-training dataset. We study the OOD performance of models pre-trained on different datasets, inlcuding ResNext101-32x8d pre-trained on ImageNet, BiTm-ResNetV2-50x3 pretrained on ImageNet-21k, and SWSL-ResNext101-32x8d pre-trained on IG-1B-Targeted. The SWSL model and BiTm model achieve similar Top-1 accuracies on ImageNet, 84.2% and 84.0%, respectively, and the standard ResNext101-32x8d achieves a slightly worse in terms of Top-1 accuracy of 79.3%. The results for this comparison are summarized in Figure 4 and Figure 15 (in Appendix).



(a) **ImageNet** pre-training. (b) **BiTm** pre-training. (c) **SWSL** pre-training.

Figure 3: Evaluating OOD generalization for three classes of models with different model sizes. **Left:** Results for ResNe(x)t models pre-trained on ImageNet-1k. **Middle:** Results for BiTm-ResNetV2 models pre-trained on ImageNet-21k. **Right:** Results for SWSL-ResNe(x)t models pre-trained on IG-1B-Targeted. For a given model class, each model size is represented by a distinct color.

Our first observation is that the pre-training dataset has a big impact on OOD generalization performance. With same architecture and model size, the SWSL pre-trained model consistently outperforms the standard ImageNet pre-trained model across. Secondly, we find that SWSL generally performs the best among all three models as it is pre-trained on the largest and the most diverse pre-training dataset. Furthermore, we find that the BiTm model pre-trained on ImageNet-21k also generally performs better than the ResNext model pre-trained on ImageNet (except for the PACS dataset, where both models have similar OOD performance).

Another interesting finding is that the OOD accuracy can be significantly improved by chosing the right fine-tuning learning rate, while the improvement in the ID accuracy is small. For example, in Figure 4(a), the ID accuracy improves from 98% to 99%, whereas the OOD accuracy increases $\sim 10\%$. Our results suggest that the "*accuracy on the line*" phenomenon [34] does not always hold for the benchmark datasets used in domain generalization. This also echoes the observation from D'Amour et al. [8], that OOD generalization can be very different among models with similar ID accuracies.

Effect of model size. It is evident from Figure 4, where both the baseline model (i.e., ResNet50) and ResNext101-32x8d are pre-trained on the ImageNet-1k dataset, that increasing the model size from ResNet50 to ResNext101-32x8d can improve OOD generalization. For example, OOD generalization is improved from 78.0% to 84.9% in Figure 4(a). Motivated by these promising results, we conduct experiments to understand the role of model size for OOD generalization. We consider increasing model size on three classes of models, including ResNe(x)t pre-trained on ImageNet-1k, BiTm-ResNetV2 pre-trained on ImageNet-21k, and ResNe(x)t pre-trained on IG-1B-Targeted. We investigate three model sizes for each class of pre-trained models and the results are summarized in Figure 3 and Table 4-6 (in Appendix).

For all three classes of models, we find that increasing the model size can improve OOD generalization in many settings. Overall, this shows that model size is a crucial design choice for improving OOD generalization. Furthermore, we note that SWSL-ResNet50 is pre-trained on a larger and more diverse dataset than BiTm-ResNetV2-50x3 and both models achieve $\sim 100\%$ training accuracy, but SWSL-ResNet50 has lower OOD accuracy than BiTm-ResNetV2-50x3 on **OfficeHome** (C). This suggests that *both* the pre-training dataset and the model size play key role in determining the OOD generalization, and ideally it is preferable to employ larger models pre-trained on larger and more diverse pre-training data.

3 Conclusion

Pre-training is one of the most effective approaches for improving model performance in a wide range of machine learning tasks. In this work, we perform an empirical study of fine-tuning a diverse set of pre-trained models and evaluate their OOD generalization. We find that simply fine-tuning *larger* models pre-trained on *more (diverse) data* can significantly improve model performance on OOD data. Additionally, and we also identify the *regularization effect of small learning rates* that is important for achieving better OOD generalization. Further, through extensive experimentation, we demonstrate that, while relying on pre-trained models, *model size* and *pre-training dataset* play a key role in ensuring good OOD generalization. We hope our results can further inspire future research on bridging the gap between in-distribution accuracy and out-of-distribution accuracy. There are multiple interesting immediate directions to explore in future work that we discuss next.

References

[1] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[3] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

[4] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1): 151–175, 2010.

[5] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[6] Dan A Calian, Florian Stimberg, Olivia Wiles, Sylvestre-Alvise Rebuffi, Andras Gyorgy, Timothy Mann, and Sven Gowal. Defending against image corruptions through adversarial augmentations. *arXiv preprint arXiv:2104.01086*, 2021.

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[8] 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. *arXiv preprint arXiv:2011.03395*, 2020.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[11] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013.

[12] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020.

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[14] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[17] Christina Heinze-Deml and Nicolai Meinshausen. Conditional variance penalties and domain shift robustness. *arXiv preprint arXiv:1710.11469*, 2017.

[18] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018.

[19] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[20] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

[21] 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. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[22] 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.

[23] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 491–507. Springer, 2020.

[24] Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

[25] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[26] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[27] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639, 2018.

[28] Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. *arXiv preprint arXiv:1907.04595*, 2019.

[29] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019.

[30] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[31] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *International Conference on Machine Learning*, pages 7313–7324. PMLR, 2021.

[32] Horia Mania and Suvrit Sra. Why do classifier accuracies show linear trends under distribution shift? *arXiv preprint arXiv:2012.15483*, 2020.

[33] 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*, pages 6905–6916. PMLR, 2020.

[34] 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*, pages 7721–7735. PMLR, 2021.

[35] Preetum Nakkiran. Learning rate annealing can provably help generalization, even for convex problems. *arXiv preprint arXiv:2005.07360*, 2020.

[36] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[37] Siyuan Qiao, Huiyu Wang, Chenxi Liu, Wei Shen, and Alan Yuille. Micro-batch training with batch-channel normalization and weight standardization. *arXiv preprint arXiv:1903.10520*, 2019.

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

[39] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

[40] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.

[41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[42] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020.

[43] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pages 9229–9248. PMLR, 2020.

[44] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[45] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

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

[47] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[48] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[49] V Vapnik. Statistical learning theory new york. *NY: Wiley*, 1:2, 1998.

[50] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[51] Georg Volk, Stefan Müller, Alexander von Bernuth, Dennis Hospach, and Oliver Bringmann. Towards robust cnn-based object detection through augmentation with synthetic rain variations. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 285–292. IEEE, 2019.

[52] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *arXiv preprint arXiv:1805.12018*, 2018.

[53] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Wenjun Zeng, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. *arXiv preprint arXiv:2103.03097*, 2021.

[54] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[55] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 819–828, 2020.

[56] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[57] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

[58] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. *Advances in Neural Information Processing Systems*, 31:8559–8570, 2018.

[59] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *International Conference on Learning Representations*, 2020.

[60] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *arXiv preprint arXiv:2103.02503*, 2021.

A Preliminaries and Experimental Setup

We begin this section by introducing the *out-of-distribution generalization* problem, with the primary focus on a special case of this broader issue, namely *domain generalization*. Subsequently, we describe the experimental setup adopted in our empirical study and various parameter choices we make throughout the paper.

In this paper, we study a general multi-class classification setting, with all input instances and their labels belonging to $\mathcal{X}$ and $\mathcal{Y} := \{1, \ldots, K\}$, respectively. Let $\mathcal{D}$ denote the training dataset which may potentially comprises data belong to k different training domains $\{\mathcal{D}^j\}_{j\in[k]:=\{1,\ldots,k\}}$, i.e., $\mathcal{D} = \cup_{j\in[k]}\mathcal{D}^j$. We assume that the training data from the j-th domain $\mathcal{D}^j = \{(\boldsymbol{x}_i^j, y_i^j)\}_{i\in[n_j]} \subset \mathcal{X} \times \mathcal{Y}$ is sampled from the distribution P_{id}^j, i.e., $(\boldsymbol{x}_i^j, y_i^j) \sim P_{\mathrm{id}}^j$. At test time, we evaluate a trained model for both its in-distribution and out-of-distribution performance. The in-distribution performance is evaluated on a test dataset $\mathcal{T}_{\mathrm{id}}$ that consists of instances belonging to the domains encountered in the training dataset $\mathcal{D}$. On the other hand, we utilize an OOD test dataset $\mathcal{T}_{\mathrm{ood}}$ from an unseen domain with distribution P_{ood} to assess the model's OOD performance.

Given a family of candidate classifiers $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}$ and the underlying training dataset $\mathcal{D}$, we primarily employ standard *empirical risk minimization* (ERM) [49] to learn a classifier $\hat{f}$ as follows:

$$\widehat{f} = \operatorname*{argmin}_{f\in\mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{i\in\mathcal{D}} \ell(f(\boldsymbol{x}_i), y_i), \tag{1}$$

where $\ell(\cdot,\cdot)$ denotes the cross-entropy loss function. For a classifier f, we define its *in-distribution accuracy* $\mathtt{Acc}_{\mathrm{id},f}$ and *out-of-distribution accuracy* $\mathtt{Acc}_{\mathrm{ood},f}$ as follows:

$$\mathtt{Acc}_{\mathrm{id},f} = \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{T}_{\mathrm{id}}}\big[\mathbb{1}\{\widehat{f}(\boldsymbol{x}) \neq y\}\big]; \quad \mathtt{Acc}_{\mathrm{ood},f} = \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{T}_{\mathrm{ood}}}\big[\mathbb{1}\{\widehat{f}(\boldsymbol{x}) \neq y\}\big], \tag{2}$$

where $\mathbb{1}\{\cdot\}$ denotes the standard indicator function. Often, models that achieve large in-distribution accuracy $\mathtt{Acc}_{\mathrm{id}}$ only achieve relatively small out-of-distribution accuracy $\mathtt{Acc}_{\mathrm{ood}}$, i.e., $\mathtt{Acc}_{\mathrm{id}} \gg \mathtt{Acc}_{\mathrm{ood}}$ [47, 19, 14]. Thus, under the domain generalization problem, one particularly focuses on designing training methods that result in classifier with good performance on both out-of-distribution data and in-distribution data.

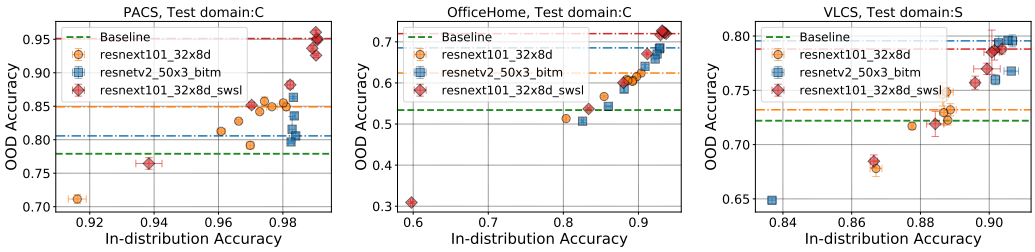
Pre-trained models. We mainly focus on fine-tuning pre-trained models on the training dataset $\mathcal{D}$ by using ERM. We explore four classes of pre-trained models for OOD generalization: (1). ResNet-based models [15, 56] pre-trained on ImageNet [41] (ResNet50, ResNext50-32x4d, and ResNext101-32x8d); (2). (BiTm)-ResNet-v2-based models [16] pre-trained on ImageNet-21k [9] (ResNetV2-50x1, ResNetV2-50x3, and ResNetV2-101x1) [23], where group normalization [54] and weight standardization [37] are used in ResNetV2; (3). (SWSL)-ResNet-based semi-weakly supervised ImageNet models pre-trained on IG-1B-Targeted data [57]; and (4). Vision transformer (ViT) pre-trained on ImageNet-21k [10]. A detailed description of these pre-trained models can be found in Table 3 (in Appendix).

OOD datasets. In our experiments, we use four vision datasets used to benchmark domain generalization algorithms [14]: (1). **PACS** dataset [25]; (2). **Office-Home** dataset [50]; (3). **VLCS** dataset [11]; and (4). **TerraIncognita** dataset [3]. Each of these datasets contains 4 different domains. We train the models on 3 domains and treat the examples from the remaining domain as the out-of-distribution data $\mathcal{T}_{\mathrm{ood}}$. For the three training domains, we use 80% data as training dataset and the remaining 20% data for evaluation. We add the test domain information after the dataset to specify the test domain, for example, **PACS** (S) means the training domains are 'P', 'A', and 'C' and the test (OOD) domain is 'S'.

Fine-tuning and model selection. We use stochastic gradient descent (SGD) with a momentum of 0.9 for fine-tuning all pre-trained models considered in this paper. The default weight decay for SGD is set to be 0. We use a cosine learning rate decay [30] as learning rate scheduler for SGD. For initial learning rates η we compare the following set $\{0.05, 0.02, 0.01, 0.005, 0.002, 0.001, 0.0005, 0.0002, 0.0001\}$. For evaluation, we pick the five checkpoints from each model with highest in-distribution accuracy. We then compute the average OOD accuracy of these five checkpoints.

Table 1: Comparison with ERM baseline results from Gulrajani and Lopez-Paz [14]. We compare our approach with the baseline in terms of the OOD accuracy $\texttt{Acc}_{\text{ood}}$. Note that our approach amounts to fine-tuning SWSL-ResNext101-32x4d [57] with different learning rates and employing the models selection procedure described in Section A. Note that, for each benchmark, we treat one of the four domains as the OOD domain and fine-tune the model on the remaining three domains. We report results for all four choices for the OOD domain on each benchmark.

	OOD Domain					OOD Domain			
PACS	A	C	P	S	**VLCS**	C	L	S	V
Baseline	88.1	78.0	97.8	79.1	Baseline	97.6	63.3	72.2	76.4
Ours	96.2	94.6	99.4	91.3	Ours	98.2	66.1	77.0	80.5
Office-Home	A	C	P	R	**TerraIncognita**	L100	L38	L43	L46
Baseline	62.7	53.4	76.5	77.3	Baseline	50.8	42.5	57.9	37.6
Ours	76.4	68.5	86.5	87.6	Ours	48.3	47.5	57.2	43.7



(a) **PACS** (C). (b) **OfficeHome** (C). (c) **VLCS** (S).

Figure 4: Evaluating OOD and ID performance of models pre-trained on different datasets. Each color corresponds to the models pre-trained on a distinct dataset and the dash-dotted line represents the model picked by our model selection procedure. For each model we report the accuracy across different fine-tuning learning rates.

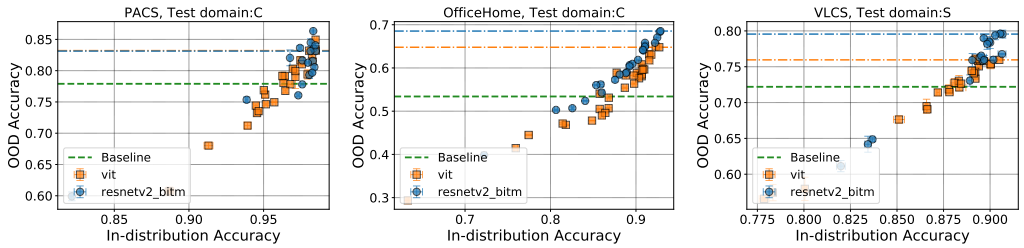
B More Experimental Results

B.1 Pre-training for better OOD generalization (Additional Results)

Effect of model architecture: ViTs vs. CNNs. We now investigate the role that the model architecture plays in ensuring good OOD generalization. Compared with convolutional neural networks (CNNs), the recently proposed Vision Transformers (ViT) achieves similar or even better performance on image classification tasks [10]. This raises the question whether vision transformers behave differently from CNNs in terms of their OOD performance? Towards this, we explore three BiT-ResNetV2 pre-trained models [23] and four ViT pre-trained models [10]. In particular, we compare BiTm-ResNetV2-$\{50\text{x}1, 101\text{x}1, 50\text{x}3\}$ with ViT-$\{$small-patch32, small-patch16, base-patch32, base-patch16$\}$. We present the comparison between ViTs and BiTms on OOD benchmarks in Figure 5 and Figure 16 (in Appendix).

We find that the OOD generalization accuracy of ViT models is similar to that of BiTm models. In some cases, BiTs slightly outperform ViTs on OOD generalization, for example, results shown in Figure 5(b), 5(c), 16(a). Since both classes of models are pre-trained on the same pre-training dataset (i.e., ImageNet-21k) and achieve similar ImageNet Top-1 accuracies (84.5% for ViTs vs. 84.0% for BiTms), our results indicate that replacing convolution operation with self-attention operation does not bring additional benefits in terms of OOD generalization for the settings we consider in this paper.

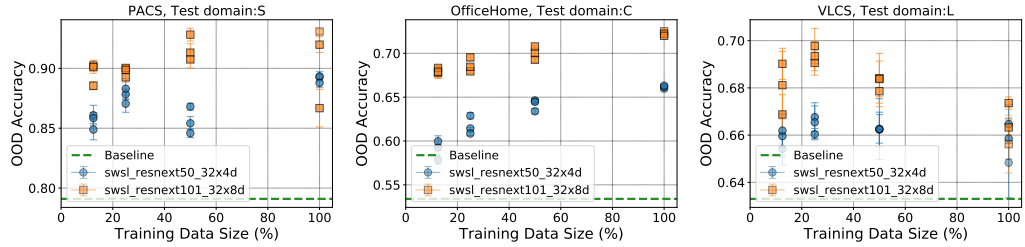
OOD performance of models trained from random initialization. To further validate the importance of pre-training data along with the model size for better OOD generalization, we train models with increasing model sizes from random initialization, i.e., without employing a pre-training phase. Our results for this experiment (cf. Table 8) show that larger models do not significantly improve the OOD generalization without the use of pre-training. For example, increasing ResNext50-32x4d to ResNext101-32x8d only improves the OOD accuracy on **OfficeHome** (C) by less than 2%. Thus, our



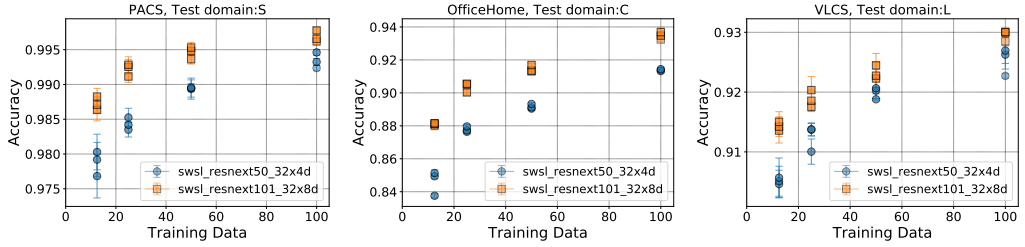
(a) **PACS** (C). (b) **OfficeHome** (C). (c) **VLCS** (S).

Figure 5: A comparison of four ViT models and three BiTm models on OOD accuracy and ID accuracy. The orange squares represent ViT models and the blue circles represent BiTm models. The dash-dotted lines represent the selected models. We do not distinguish the model architectures within the same model class.

results suggest that without pre-training, only increasing model size is not very effective in improving the OOD performance.



(a) **PACS-OOD** (S). (b) **OfficeHome-OOD** (C). (c) **VLCS-OOD** (L).



(d) **PACS-ID** (S). (e) **OfficeHome-ID** (C). (f) **VLCS-ID** (L).

Figure 6: Evaluating OOD generalization performance of models trained with different number of training samples. X-axis represents the number of training samples. We use SWSL-ResNext50-32x4d and SWSL-ResNext101-32x8d as the pre-trained models. For each pre-trained model, we visualize the OOD accuracies of the top-3 models selected by ID accuracy.

B.2 Analysis of techniques used for improving ID accuracy

In this subsection, we examine various techniques that have been shown to improve ID accuracy to assess their utility in achieving good OOD generalization. First, we study the impact of training data size on the OOD generalization, as increasing the number of training samples is an effective approach to improve ID generalization. Then we evaluate four techniques in our OOD setting, including label smoothing [45], AutoAugment [7], PatchGaussian [29], and Sharpness-Aware Minimization (SAM) [12].

Utility of more training data. We consider fine-tuning with datsets of four different sizes, i.e., $100\%, 50\%, 25\%,$ and 12.5% of the total training data for a given benchmark. Our results in Figure 6 show that increasing the training data from 12.5% to 100% does not significantly improve OOD generalization. In fact, on **VLCS** (S), a better OOD generalization is realized when we utilize less training data to train a ResNext101-32x8d model (cf. Figure 6(c)). This suggests that increasing ID training samples is not as effective as using larger and better pre-trained models.

Table 2: Evaluation of four techniques (label smoothing, AutoAugment, PatchGaussian, and SAM) for OOD generalization. We use the same pre-trained model (SWSL-ResNext101-32x4d) across all settings. The number inside the parentheses after the method name represents the value of the technique-specific hyperparameter, e.g., PatchGaussian (1.0) corresponds to employing PatchGaussian [29] with $\sigma = 1.0$. We highlight the best two OOD accuracies for each dataset with bold text.

Method	**PACS** (C)	**Office** (C)	**VLCS** (L)	**Terra** (L46)
ERM (in Table 1)	**94.6**	68.5	66.1	43.7
Label Smoothing (0.1)	91.6	70.6	65.2	42.5
Label Smoothing (0.2)	93.5	70.8	66.3	**44.6**
AutoAugment	93.5	70.7	65.2	38.1
PatchGaussian (1.0)	92.8	65.8	64.6	13.1
PatchGaussian (0.5)	**94.4**	69.3	63.6	9.7
SAM (0.02)	93.5	**72.2**	**67.1**	**44.7**
SAM (0.05)	92.8	**71.3**	**67.5**	42.3

Methods for improving ID accuracy. We find that applying the methods listed in Table 2 does not significantly improve the OOD generalization across four datasets, and utilizing augmentations/regularization can potentially even hurt OOD generalization. For example, applying PatchGaussian decreases OOD accuracy on **Terra** (L46) from 43.7% to 13.1% and 9.7% with $\sigma = 1.0$ and 0.5, respectively. In constrast, PatchGaussian has very minimal impact on the ID performance, where it achieves $\texttt{Acc}_{\text{id}} = 95.8\%$ and $\texttt{Acc}_{\text{id}} = 95.6\%$, respectively; ERM attains $\texttt{Acc}_{\text{id}} = 95.9\%$. Contrary to our results, Lopes et al. [29] notice that PatchGaussian can improve both the clean (ID) accuracy and robustness to common corruptions (OOD accuracy). This suggests that one should employing augmentation/regularization techniques carefully so as to not harm OOD generalization as a side effect. On the other hand, SAM with parameter[1] $\rho = 0.02$ improves the OOD generalization on three benchmarks. Overall, we find that methods used for improving ID accuracy do not necessarily improve OOD accuracy, when compared with the simple ERM-based fine-tuning approach.

C Related Work

Domain adaptation and domain generalization. Ben-David et al. [4] proposed $\mathcal{H}\Delta\mathcal{H}$-divergence and applied it to develop error bounds on new test data distribution that are different from the training distribution. Motivated by the theoretical results in Ben-David et al. [4], a large body of work was devoted to learning domain-invariant representations for domain adaptation where unlabeled data from the target domain are available during training [1, 13, 48, 58]. Different from the domain adaptation problem, no information from the target domain is available in the domain generalization problem. Recent works proposed various algorithms to improve domain generalization, including learning invariant representations across training domains [27, 2], distributionally robust optimization Sagawa et al. [42], data augmentation [52, 59], and causal framework [17, 31]. Zhou et al. [60], Wang et al. [53] provide a more detailed review of the topic of domain generalization.

OOD generalization and model robustness. Recent works developed new datasets to evaluate model robustness to out-of-distribution data, including ImageNet-V2 [40], CIFAR-10.1 [39], ImageNet-C and CIFAR-10-C [18], ImageNet-R [20], WILDS [22], etc. A line of work proposed methods to improve model robustness to commons corruptions [29, 55, 21, 6]. On multiple datasets, the researchers have observed a near linear correlation between the OOD accuracy and the ID accuracy [40, 33, 32, 34]. However, Taori et al. [46] found that most approaches, including the ones that improve robustness to synthetic distribution shifts, do not improve model robustness to natural distribution shifts, except for the models that are pre-trained on larger datasets. Similar observation has been made in Hendrycks et al. [20]. Our work focuses on the domain generalization benchmark datasets as well as how to perform fine-tuning one various pre-trained models for better OOD generalization.

[1] The perturbation parameter ρ is defined in [12].

D Discussion and Future Work

Scaling model size and pre-training dataset size. Our empirical results indicate that model size and the pre-training dataset size are two essential factors for improving OOD performance. Even though we primarily focus on image classification benchmarks, similar behavior has been observed in NLP domain [5], where increasing model size leads to monotonic improvements in zero-shot performance on unseen tasks for example. It is worth noting that while increasing model size or pre-training dataset size only marginally improves the in-distribution accuracy, the increase in OOD accuracy can be much larger. It is an interesting direction to explore the differences in ID and OOD generalization in other domains.

How to perform model selection? Our study indicates that models trained via different methods can exhibit a large variation in terms of their OOD performance, even when their in-distribution accuracy is very similar, e.g., see Figure 4(c) and Figure 5(c). We thus believe that the development of better model selection approaches for OOD generalization will be a key direction of future work.

Limitations of ERM. Despite impressive results for OOD generalization, we find that ERM-based fine-tuning of pre-trained models is unlikely to close the in-distribution to out-of-distribution generalization gap , even when the model size or the pre-training dataset becomes much larger. To generalize to unseen domains, it might be necessary to have access to extra information about the OOD dataset (e.g., unlabeled OOD data as available in the domain adaptation setting [36]) and/or design novel algorithms (e.g., updating the model parameters during test time [43]).

E Additional Results

E.1 Additional Experimental Details

Implementation details. Our implementation is mainly adapted from `DomainBed` (developed in Gulrajani and Lopez-Paz [14]). Before we fine-tune the pre-trained models, we replace the final linear layer of the pre-trained model with a random initialized linear layer with output dimension equals to the number of classes for the dataset. Most of the pre-trained models are from Py**T**orch **Im**age **M**odels (`timm`). The default training iteration is $5,000$. For the training from scratch setting (i.e., without pre-training), we set the training iteration as $10,000$ and weight decay as 10^{-5}. During the fine-tuning, we save model checkpoint every 100 iterations and select five checkpoints that achieves the top-5 in-distribution accuracies. The input size of the ViT models is $(3, 224, 224)$. In Table 3, we summarize the top-1/top-5 ImageNet accuracies of pre-trained models used in this paper.

Table 3: Top-1 accuracy and Top-5 accuracy of pre-trained models considered in this paper. (**ImageNet**) ResNet-based models pre-trained on ImageNet-1k. (**BiTm**) ResNetV2-based models pre-trained on ImageNet-21k. (**SWSL**) ResNet-based models pre-trained on IG-1B-Targeted. (**ViT**) Vision transformer-based models pre-trained on ImageNet-21k.

ImageNet	ResNet50	ResNext50-32x4d	ResNext101-32x8d	
Top-1 `Acc`	76.13	77.62	79.30	
Top-5 `Acc`	92.86	93.69	94.51	
BiTm	ResNetV2-50x1	ResNetV2-101x1	ResNetV2-50x3	
Top-1 `Acc`	80.34	82.33	84.01	
Top-5 `Acc`	95.68	96.51	97.12	
SWSL	ResNet50	ResNext50-32x4d	ResNext101-32x4d	ResNext101-32x8d
Top-1 `Acc`	81.16	82.18	83.23	84.28
Top-5 `Acc`	95.97	96.23	96.76	97.17
ViT	Small-Patch32	Base-Patch32	Small-Patch16	Base-Patch16
Top-1 `Acc`	75.99	80.72	81.40	84.53
Top-5 `Acc`	93.27	95.56	96.13	97.29

E.2 Additional Experimental Results

In this subsection, we present additional experimental results in Section 2.

Effect of fine-tune learning rate in Section 2.2. In addition to Figure 2 and Figure 13, we provide more experimental results on the effect of fine-tune learning rate. For fine-tuning results with different learning rates, we present more results for fine-tuning SWSL-ResNext101-32x4d in Figure 7, results for fine-tuning SWSL-ResNext101-32x8d in Figure 8, results for fine-tuning SWSL-ResNext50-32x4d in Figure 9, results for fine-tuning BiTm-ResNetV2-50x3 in Figure 10, results for fine-tuning ResNext101-32x8d (ImageNet pre-trained) in Figure 11, and results for fine-tuning ResNext50-32x4d (ImageNet pre-trained) in Figure 12. Meanwhile, in Figure E.2, we provide more results on visualizing the OOD accuracy vs. training loss for SWSL-ResNext101-32x4d.

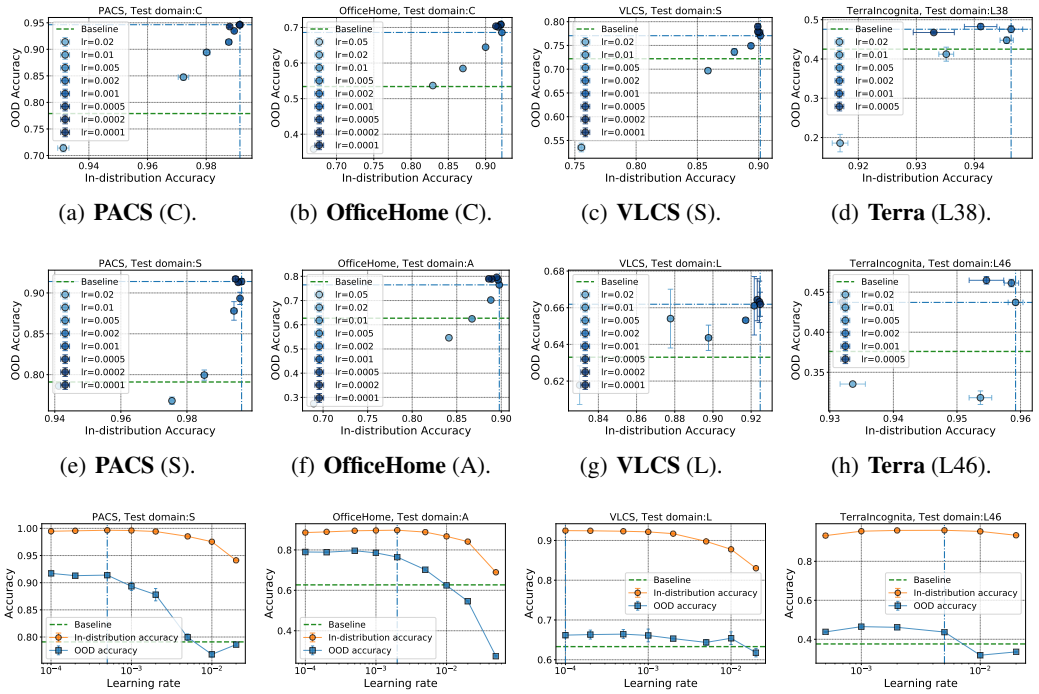
Effect of pre-training dataset. In Figure 15, we provide additional experimental results on studying the effect of pre-training dataset.

Effect of model architecture: ViTs vs. CNNs. In Figure 16, we provide additional experimental results on comparing the performance of ViTs and CNNs.

Effect of model size. We provide additional results on comparing models with different model sizes in Table 4 (ImageNet-1k pre-trained models), Table 5 (BiTm pre-trained models [23]), Table 6 (SWSL pre-trained models [57]), Table 7 (ViT pre-trained models [10]), and Table 8 (without pre-training).

Utility of more training data. In Figure 17, we provide additional experimental results on investigating the impact of the number of training samples on the **TerraIncognita** dataset, including the ID/OOD accuracy results.

Methods for improving ID accuracy. In Table 9, we provide the in-distribution accuracy evaluations of the methods described in Table 2. Also, in Table 10 and 11, we provide additional results on the ID/OOD accuracy evaluations of different methods (on more datasets).

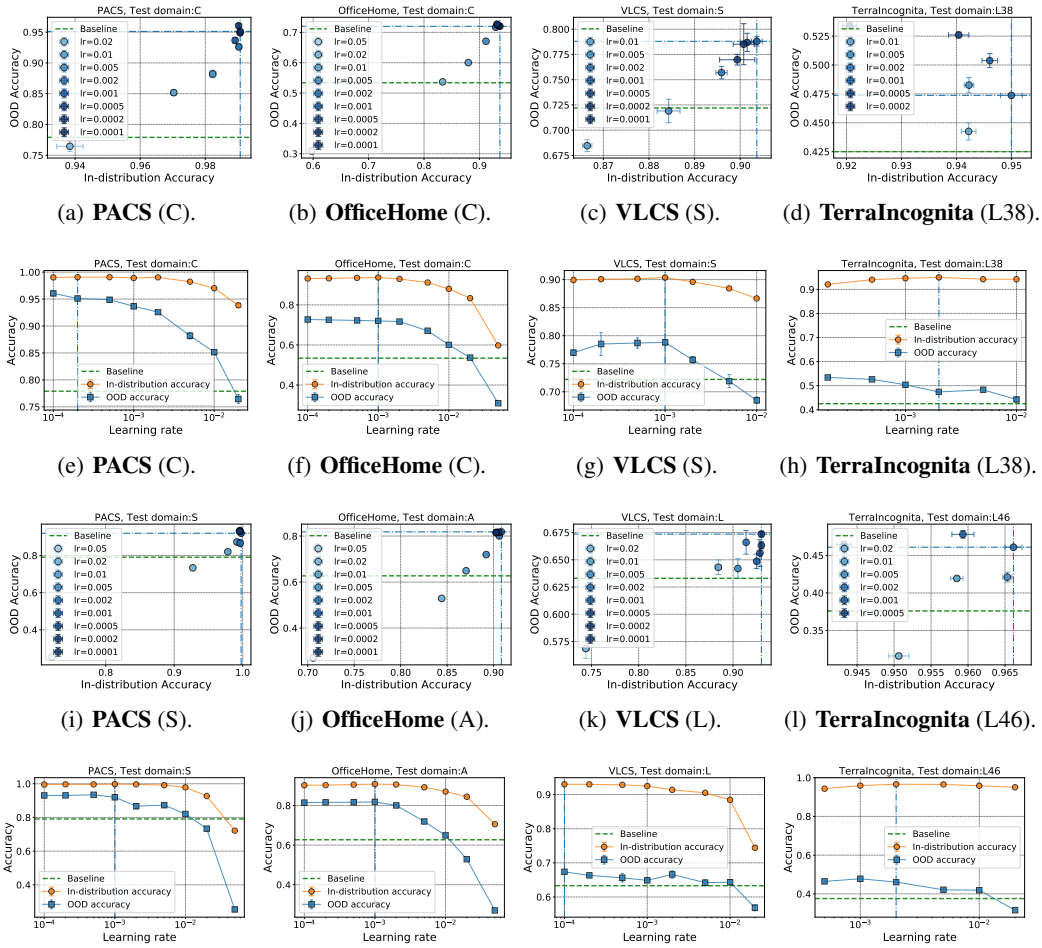


(a) **PACS** (C). (b) **OfficeHome** (C). (c) **VLCS** (S). (d) **Terra** (L38).

(e) **PACS** (S). (f) **OfficeHome** (A). (g) **VLCS** (L). (h) **Terra** (L46).

(i) **PACS** (S). (j) **OfficeHome** (A). (k) **VLCS** (L). (l) **Terra** (L46).

Figure 7: (Additional results) Evaluating models (SWSL-ResNext101-32x4d) fine-tuned by different learning rates on in-distribution and out-of-distribution data. (Top row) Scatter plot of in-distribution accuracy (X-axis) and out-of-distribution accuracy (Y-axis). (Bottom row) Compare in-distribution accuracy with out-of-distribution accuracy w.r.t. learning rate (X-axis). The green dashed line corresponds to the baseline OOD accuracy, and the blue dash-dotted line represents the selected model (by selecting the model with best in-distribution accuracy).



(a) **PACS** (C). (b) **OfficeHome** (C). (c) **VLCS** (S). (d) **TerraIncognita** (L38).

(e) **PACS** (C). (f) **OfficeHome** (C). (g) **VLCS** (S). (h) **TerraIncognita** (L38).

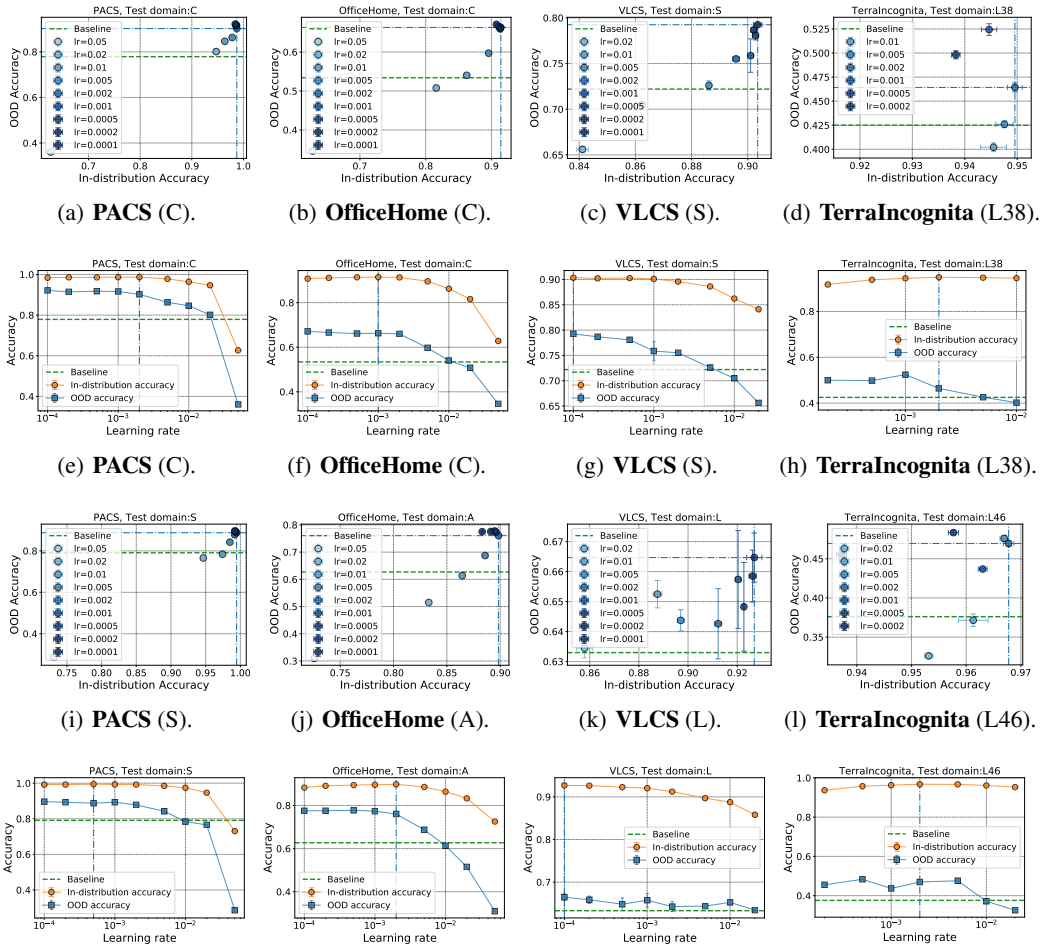
(i) **PACS** (S). (j) **OfficeHome** (A). (k) **VLCS** (L). (l) **TerraIncognita** (L46).

(m) **PACS** (S). (n) **OfficeHome** (A). (o) **VLCS** (L). (p) **TerraIncognita** (L46).

Figure 8: Evaluating models (SWSL-ResNext101-32x8d) fine-tuned by different learning rates on in-distribution and out-of-distribution data. (Top row) Scatter plot of in-distribution accuracy (X-axis) and out-of-distribution accuracy (Y-axis). (Bottom row) Compare in-distribution accuracy with out-of-distribution accuracy w.r.t. learning rate (X-axis). The green dashed line corresponds to the baseline OOD accuracy, and the blue dash-dot line represents the selected model (by selecting the model with best in-distribution accuracy).

Table 4: Comparing ImageNet pre-trained models with different model sizes. We evaluate both in-distribution accuracy and out-of-distribution. (*left*: ID accuracy, *right*: OOD accuracy).

Models	**PACS** (C)	**PACS** (S)
ResNet50	0.977 ± 0.000 / 0.785 ± 0.006	0.980 ± 0.001 / 0.698 ± 0.008
ResNext50-32x4d	0.977 ± 0.001 / 0.825 ± 0.005	0.982 ± 0.001 / 0.698 ± 0.002
ResNext101-32x8d	0.981 ± 0.000 / 0.849 ± 0.005	0.988 ± 0.000 / 0.765 ± 0.013
Models	**Office-Home** (A)	**Office-Home** (C)
ResNet50	0.869 ± 0.001 / 0.662 ± 0.002	0.874 ± 0.002 / 0.542 ± 0.003
ResNext50-32x4d	0.874 ± 0.000 / 0.645 ± 0.003	0.887 ± 0.002 / 0.558 ± 0.003
ResNext101-32x8d	0.887 ± 0.000 / 0.717 ± 0.002	0.903 ± 0.000 / 0.624 ± 0.003
Models	**VLCS** (L)	**VLCS** (S)
ResNet50	0.898 ± 0.000 / 0.645 ± 0.003	0.885 ± 0.001 / 0.745 ± 0.003
ResNext50-32x4d	0.905 ± 0.000 / 0.646 ± 0.007	0.891 ± 0.002 / 0.733 ± 0.003
ResNext101-32x8d	0.909 ± 0.001 / 0.643 ± 0.002	0.888 ± 0.001 / 0.732 ± 0.005



(a) **PACS** (C). (b) **OfficeHome** (C). (c) **VLCS** (S). (d) **TerraIncognita** (L38).

(e) **PACS** (C). (f) **OfficeHome** (C). (g) **VLCS** (S). (h) **TerraIncognita** (L38).

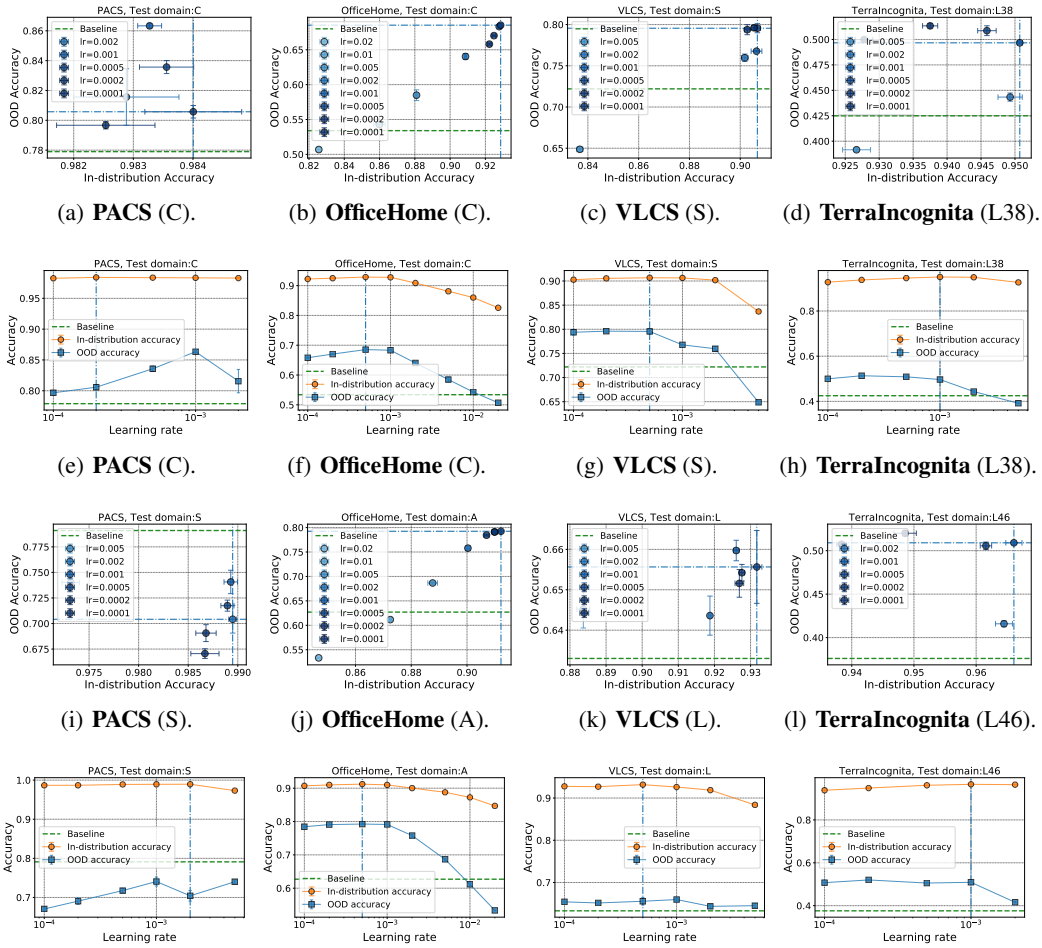
(i) **PACS** (S). (j) **OfficeHome** (A). (k) **VLCS** (L). (l) **TerraIncognita** (L46).

(m) **PACS** (S). (n) **OfficeHome** (A). (o) **VLCS** (L). (p) **TerraIncognita** (L46).

Figure 9: Evaluating models (SWSL-ResNext50-32x4d) fine-tuned by different learning rates on in-distribution and out-of-distribution data. (Top row) Scatter plot of in-distribution accuracy (X-axis) and out-of-distribution accuracy (Y-axis). (Bottom row) Compare in-distribution accuracy with out-of-distribution accuracy w.r.t. learning rate (X-axis). The green dashed line corresponds to the baseline OOD accuracy, and the blue dash-dot line represents the selected model (by selecting the model with best in-distribution accuracy).

Table 5: Comparing BiTm models with different model sizes. We evaluate both in-distribution accuracy and out-of-distribution. (*left*: ID accuracy, *right*: OOD accuracy).

Models	**PACS** (C)	**PACS** (S)
BiTm-ResNetV2-50x1	0.980 ± 0.001 / 0.793 ± 0.005	0.983 ± 0.001 / 0.618 ± 0.002
BiTm-ResNetV2-101x1	0.984 ± 0.000 / 0.831 ± 0.002	0.990 ± 0.001 / 0.669 ± 0.007
BiTm-ResNetV2-50x3	0.984 ± 0.000 / 0.805 ± 0.004	0.989 ± 0.000 / 0.704 ± 0.013
Models	**Office-Home** (A)	**Office-Home** (C)
BiTm-ResNetV2-50x1	0.887 ± 0.001 / 0.721 ± 0.001	0.895 ± 0.001 / 0.609 ± 0.002
BiTm-ResNetV2-101x1	0.904 ± 0.001 / 0.756 ± 0.001	0.910 ± 0.000 / 0.651 ± 0.000
BiTm-ResNetV2-50x3	0.912 ± 0.000 / 0.792 ± 0.001	0.928 ± 0.000 / 0.685 ± 0.002
Models	**VLCS** (L)	**VLCS** (S)
BiTm-ResNetV2-50x1	0.922 ± 0.000 / 0.647 ± 0.010	0.896 ± 0.000 / 0.759 ± 0.006
BiTm-ResNetV2-101x1	0.923 ± 0.002 / 0.656 ± 0.001	0.900 ± 0.000 / 0.783 ± 0.008
BiTm-ResNetV2-50x3	0.931 ± 0.001 / 0.655 ± 0.009	0.906 ± 0.001 / 0.795 ± 0.005



(a) **PACS** (C). (b) **OfficeHome** (C). (c) **VLCS** (S). (d) **TerraIncognita** (L38).

(e) **PACS** (C). (f) **OfficeHome** (C). (g) **VLCS** (S). (h) **TerraIncognita** (L38).

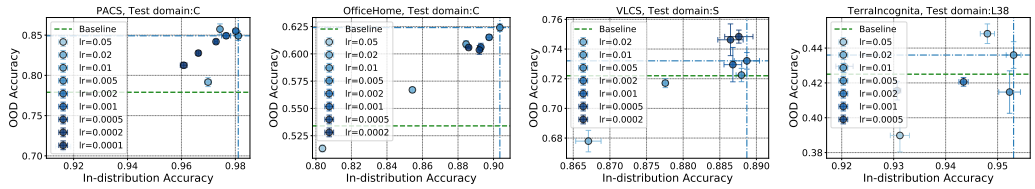
(i) **PACS** (S). (j) **OfficeHome** (A). (k) **VLCS** (L). (l) **TerraIncognita** (L46).

(m) **PACS** (S). (n) **OfficeHome** (A). (o) **VLCS** (L). (p) **TerraIncognita** (L46).

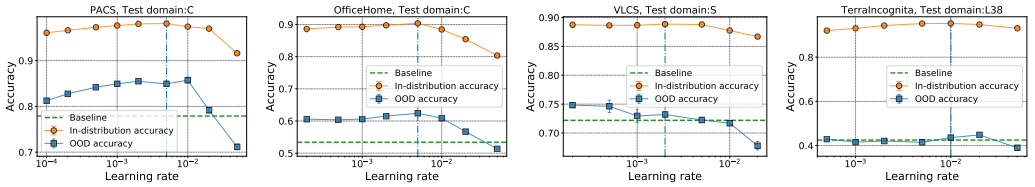
Figure 10: Evaluating BiTm models (BiTm-ResNetV2-50x3) fine-tuned by different learning rates on in-distribution and out-of-distribution data. (Top row) Scatter plot of in-distribution accuracy (X-axis) and out-of-distribution accuracy (Y-axis). (Bottom row) Compare in-distribution accuracy with out-of-distribution accuracy w.r.t. learning rate (X-axis). The green dashed line corresponds to the baseline OOD accuracy, and the blue dash-dot line represents the selected model (by selecting the model with best in-distribution accuracy).

Table 6: Comparing SWSL models with different model sizes. We evaluate both in-distribution accuracy and out-of-distribution. (*left*: ID accuracy, *right*: OOD accuracy).

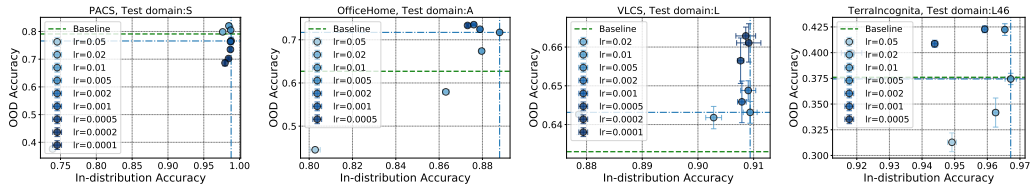
Models	**PACS** (C)	**PACS** (S)
SWSL-ResNet50	0.987 ± 0.000 / 0.870 ± 0.008	0.992 ± 0.000 / 0.856 ± 0.005
SWSL-ResNext50-32x4d	0.987 ± 0.000 / 0.902 ± 0.002	0.994 ± 0.000 / 0.887 ± 0.003
SWSL-ResNext101-32x8d	0.990 ± 0.000 / 0.951 ± 0.002	0.997 ± 0.000 / 0.919 ± 0.006
Models	**Office-Home** (A)	**Office-Home** (C)
SWSL-ResNet50	0.886 ± 0.000 / 0.729 ± 0.003	0.908 ± 0.001 / 0.658 ± 0.002
SWSL-ResNext50-32x4d	0.898 ± 0.000 / 0.760 ± 0.001	0.914 ± 0.001 / 0.663 ± 0.001
SWSL-ResNext101-32x8d	0.908 ± 0.000 / 0.818 ± 0.001	0.936 ± 0.002 / 0.719 ± 0.002
Models	**VLCS** (L)	**VLCS** (S)
SWSL-ResNet50	0.920 ± 0.000 / 0.662 ± 0.006	0.900 ± 0.000 / 0.782 ± 0.003
SWSL-ResNext50-32x4d	0.926 ± 0.003 / 0.664 ± 0.008	0.903 ± 0.001 / 0.792 ± 0.001
SWSL-ResNext101-32x8d	0.930 ± 0.000 / 0.673 ± 0.002	0.903 ± 0.001 / 0.787 ± 0.004



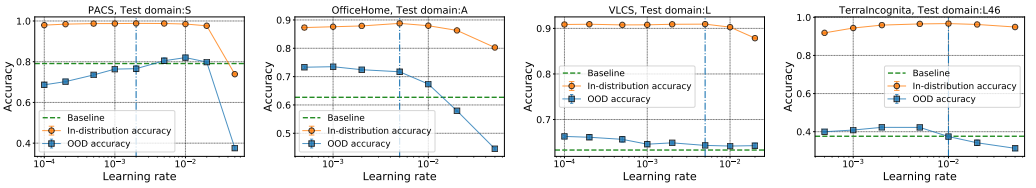
(a) **PACS** (C). (b) **OfficeHome** (C). (c) **VLCS** (S). (d) **TerraIncognita** (L38).



(e) **PACS** (C). (f) **OfficeHome** (C). (g) **VLCS** (S). (h) **TerraIncognita** (L38).

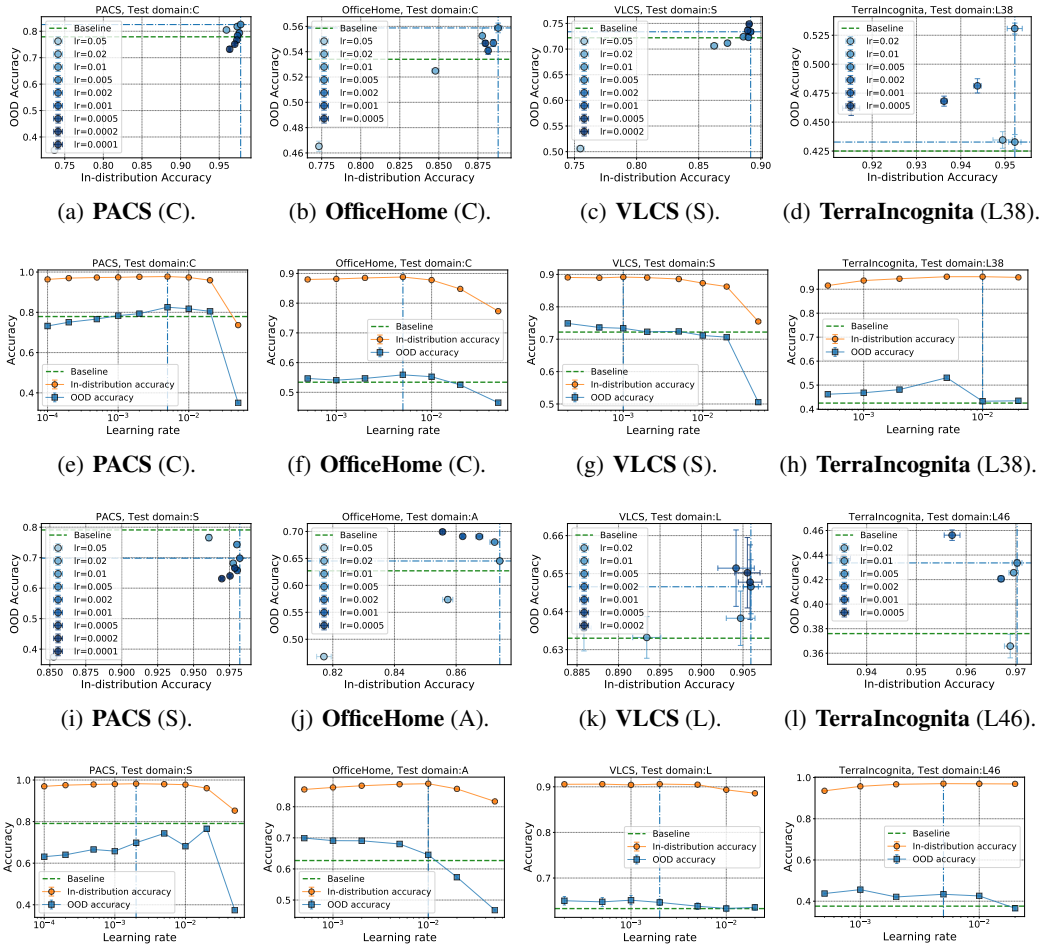


(i) **PACS** (S). (j) **OfficeHome** (A). (k) **VLCS** (L). (l) **TerraIncognita** (L46).



(m) **PACS** (S). (n) **OfficeHome** (A). (o) **VLCS** (L). (p) **TerraIncognita** (L46).

Figure 11: Evaluating models (ResNext101-32x8d pre-trained on ImageNet) fine-tuned by different learning rates on in-distribution and out-of-distribution data. (Top row) Scatter plot of in-distribution accuracy (X-axis) and out-of-distribution accuracy (Y-axis). (Bottom row) Compare in-distribution accuracy with out-of-distribution accuracy w.r.t. learning rate (X-axis). The green dashed line corresponds to the baseline OOD accuracy, and the blue dash-dot line represents the selected model (by selecting the model with best in-distribution accuracy).



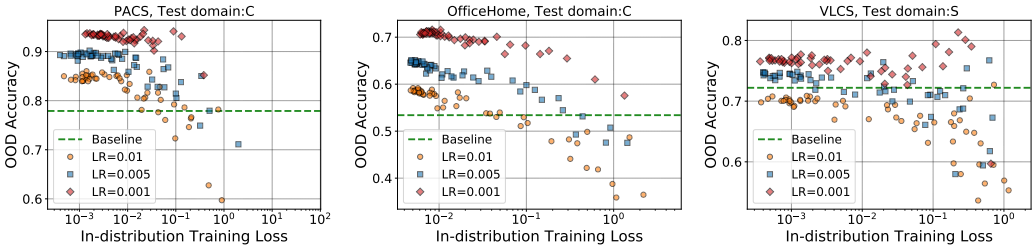
(a) **PACS** (C). (b) **OfficeHome** (C). (c) **VLCS** (S). (d) **TerraIncognita** (L38).

(e) **PACS** (C). (f) **OfficeHome** (C). (g) **VLCS** (S). (h) **TerraIncognita** (L38).

(i) **PACS** (S). (j) **OfficeHome** (A). (k) **VLCS** (L). (l) **TerraIncognita** (L46).

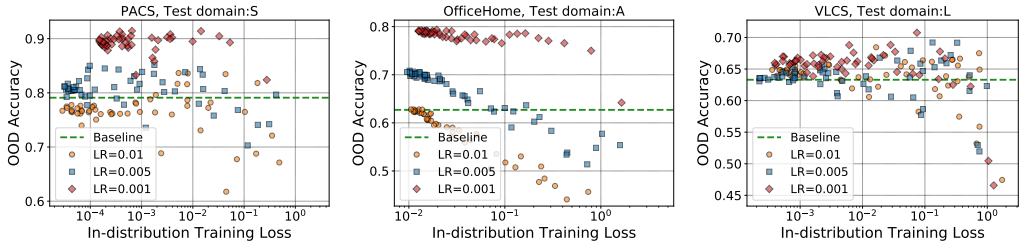
(m) **PACS** (S). (n) **OfficeHome** (A). (o) **VLCS** (L). (p) **TerraIncognita** (L46).

Figure 12: Evaluating models (ResNext50-32x4d pre-trained on ImageNet) fine-tuned by different learning rates on in-distribution and out-of-distribution data. (Top row) Scatter plot of in-distribution accuracy (X-axis) and out-of-distribution accuracy (Y-axis). (Bottom row) Compare in-distribution accuracy with out-of-distribution accuracy w.r.t. learning rate (X-axis). The green dashed line corresponds to the baseline OOD accuracy, and the blue dash-dot line represents the selected model (by selecting the model with best in-distribution accuracy).



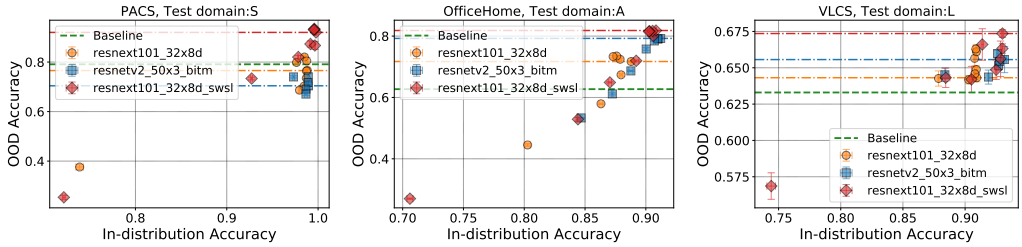
(a) **PACS** (C). (b) **OfficeHome** (C). (c) **VLCS** (S).

Figure 13: OOD accuracy of models (SWSL-ResNext101-32x4d) during training. We visualize models trained with three different learning rates in terms of OOD accuracy vs. training loss. Each point in the above plots represents the model evaluated at one iteration during training



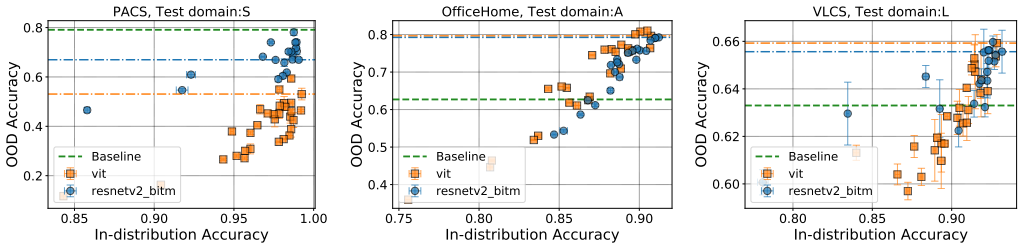
(a) **PACS** (S). (b) **OfficeHome** (A). (c) **VLCS** (L).

Figure 14: OOD accuracy of models (SWSL-ResNext101-32x4d) during training. We visualize models trained with three different learning rates in terms of OOD accuracy v.s. training loss. Each point in the above plots represents the model evaluated at one iteration during training.

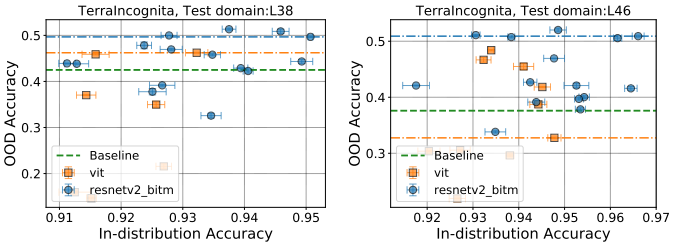


(a) **PACS** (S). (b) **OfficeHome** (A). (c) **VLCS** (L).

Figure 15: Evaluating out-of-distribution and in-distribution performance of models pre-trained on different datasets. Each color corresponds to models pre-trained on one dataset and the dash-dot line represents the selected model.



(a) **PACS** (S). (b) **OfficeHome** (A). (c) **VLCS** (L).



(d) **TerraIncognita** (L38). (e) **TerraIncognita** (L46).

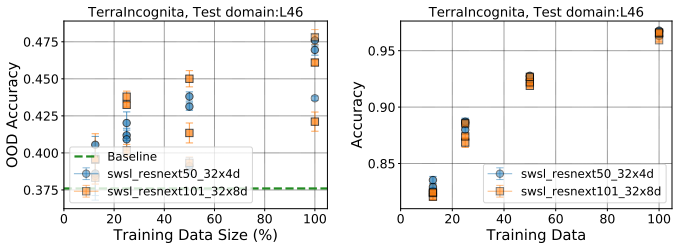
Figure 16: A comparison of four ViT models and three BiTm models on out-of-distribution accuracy and in-distribution accuracy. The orange squares represent ViT models and the blue circles represent BiTm models. The dash-dot lines represent the selected models. We do not distinguish the model architectures within the same model class.

Table 7: Comparing ViTs models with different model sizes. We evaluate both in-distribution accuracy and out-of-distribution. (*left*: ID accuracy, *right*: OOD accuracy).

Models	**PACS** (C)	**PACS** (S)
ViT-small-patch32	0.951 ± 0.001 / 0.746 ± 0.001	0.964 ± 0.001 / 0.404 ± 0.005
ViT-small-patch16	0.982 ± 0.000 / 0.825 ± 0.005	0.985 ± 0.000 / 0.494 ± 0.004
ViT-base-patch32	0.971 ± 0.000 / 0.789 ± 0.018	0.982 ± 0.000 / 0.478 ± 0.013
ViT-base-patch16	0.985 ± 0.000 / 0.832 ± 0.001	0.992 ± 0.000 / 0.530 ± 0.023
Models	**Office-Home** (A)	**Office-Home** (C)
ViT-small-patch32	0.856 ± 0.001 / 0.618 ± 0.001	0.867 ± 0.000 / 0.506 ± 0.001
ViT-small-patch16	0.888 ± 0.001 / 0.772 ± 0.001	0.907 ± 0.001 / 0.584 ± 0.003
ViT-base-patch32	0.895 ± 0.001 / 0.758 ± 0.004	0.909 ± 0.001 / 0.596 ± 0.001
ViT-base-patch16	0.907 ± 0.000 / 0.796 ± 0.001	0.927 ± 0.001 / 0.647 ± 0.001
Models	**VLCS** (L)	**VLCS** (S)
ViT-small-patch32	0.897 ± 0.001 / 0.628 ± 0.001	0.883 ± 0.003 / 0.721 ± 0.004
ViT-small-patch16	0.918 ± 0.001 / 0.643 ± 0.015	0.894 ± 0.001 / 0.759 ± 0.004
ViT-base-patch32	0.910 ± 0.001 / 0.631 ± 0.005	0.892 ± 0.002 / 0.733 ± 0.001
ViT-base-patch16	0.928 ± 0.001 / 0.659 ± 0.003	0.904 ± 0.000 / 0.759 ± 0.000

Table 8: Comparing models trained from scratch with different model sizes. We evaluate both in-distribution accuracy and out-of-distribution. (*left*: ID accuracy, *right*: OOD accuracy).

Models	**PACS** (C)	**PACS** (S)
ResNet50	0.812 ± 0.003 / 0.470 ± 0.005	0.817 ± 0.005 / 0.294 ± 0.004
ResNext50-32x4d	0.816 ± 0.003 / 0.497 ± 0.004	0.785 ± 0.001 / 0.192 ± 0.011
ResNext101-32x8d	0.681 ± 0.004 / 0.505 ± 0.010	0.770 ± 0.002 / 0.287 ± 0.007
Models	**Office-Home** (A)	**Office-Home** (S)
ResNet50	0.643 ± 0.002 / 0.201 ± 0.001	0.557 ± 0.004 / 0.226 ± 0.004
ResNext50-32x4d	0.656 ± 0.000 / 0.225 ± 0.001	0.560 ± 0.002 / 0.233 ± 0.001
ResNext101-32x8d	0.654 ± 0.002 / 0.216 ± 0.001	0.575 ± 0.001 / 0.249 ± 0.001
Models	**VLCS** (L)	**VLCS** (S)
ResNet50	0.758 ± 0.003 / 0.570 ± 0.012	0.747 ± 0.001 / 0.509 ± 0.000
ResNext50-32x4d	0.757 ± 0.000 / 0.567 ± 0.013	0.751 ± 0.001 / 0.513 ± 0.005
ResNext101-32x8d	0.760 ± 0.002 / 0.567 ± 0.004	0.739 ± 0.003 / 0.502 ± 0.002



(a) **Terra-OOD** (L46). (b) **Terra-ID** (L46).

Figure 17: Evaluating OOD generalization performance of models trained with different number of training samples on the **TerraIncognita** dataset. X-axis represents the number of training samples. We use SWSL-ResNext50-32x4d and SWSL-ResNext101-32x8d as the pre-trained models. For each pre-trained model, we visualize the OOD accuracies of the top-3 models selected by ID accuracy.

Table 9: *In-discribution* accuracy evaluation of four techniques (label smoothing, AutoAugment, PatchGaussian, and SAM) for OOD generalization. We use the same pre-trained model (SWSL-ResNext101-32x4d) across all settings. The number inside the parentheses after the method name represents the value of the technique-specific hyperparameter, e.g., PatchGaussian (1.0) corresponds to employing PatchGaussian [29] with $\sigma = 1.0$.

Method	**PACS** (C)	**Office** (C)	**VLCS** (L)	**Terra** (L46)
ERM (in Table 1)	99.1	92.1	92.4	95.9
Label Smoothing (0.1)	99.0	92.4	92.9	95.8
Label Smoothing (0.2)	99.0	91.8	93.1	95.8
AutoAugment	98.4	90.6	91.9	93.8
PatchGaussian (1.0)	99.1	92.9	92.6	95.8
PatchGaussian (0.5)	99.0	92.3	92.6	95.6
SAM (0.02)	99.2	92.9	93.3	96.1
SAM (0.05)	99.0	93.2	93.0	95.1

Table 10: *Out-of-discribution* accuracy evaluation of four techniques (label smoothing, AutoAugment, PatchGaussian, and SAM) for OOD generalization. We use the same pre-trained model (SWSL-ResNext101-32x4d) across all settings. The number inside the parentheses after the method name represents the value of the technique-specific hyperparameter, e.g., PatchGaussian (1.0) corresponds to employing PatchGaussian [29] with $\sigma = 1.0$. We highlight the best two OOD accuracies for each dataset with bold text.

Method	**PACS** (S)	**Office** (A)	**VLCS** (S)	**Terra** (L38)
ERM (in Table 1)	91.3	76.4	77.0	47.5
Label Smoothing (0.1)	**91.7**	78.0	78.0	**48.3**
Label Smoothing (0.2)	**91.7**	78.6	78.6	**50.3**
AutoAugment	91.3	78.1	77.1	46.9
PatchGaussian (1.0)	90.3	66.4	71.6	12.4
PatchGaussian (0.5)	90.0	73.5	75.9	6.2
SAM (0.02)	89.7	**80.3**	**79.2**	42.0
SAM (0.05)	90.7	**80.4**	**79.7**	43.9

Table 11: *In-discribution* accuracy evaluation of four techniques (label smoothing, AutoAugment, PatchGaussian, and SAM) for OOD generalization. We use the same pre-trained model (SWSL-ResNext101-32x4d) across all settings. The number inside the parentheses after the method name represents the value of the technique-specific hyperparameter, e.g., PatchGaussian (1.0) corresponds to employing PatchGaussian [29] with $\sigma = 1.0$.

Method	**PACS** (S)	**Office** (A)	**VLCS** (S)	**Terra** (L38)
ERM (in Table 1)	99.6	89.7	90.1	94.6
Label Smoothing (0.1)	99.5	89.8	90.6	94.8
Label Smoothing (0.2)	99.5	89.8	90.5	94.5
AutoAugment	99.6	88.6	89.9	92.2
PatchGaussian (1.0)	99.6	90.1	90.1	94.0
PatchGaussian (0.5)	99.6	90.2	90.3	94.1
SAM (0.02)	99.2	91.0	90.9	94.5
SAM (0.05)	99.7	91.2	90.9	93.3