Pre-train, fine-tune, interpolate: a three-stage strategy for domain generalization

Anonymous Author(s)
Affiliation
Address
email

Abstract

The goal of domain generalization is to train models that generalize well to unseen domains. To this end, the typical strategy is two-stage: first pre-training the network on a large corpus, then fine-tuning on the task's training domains. If the pre-training dataset is large enough, this pre-training is efficient because it will contain samples related to the unseen domains. Yet, large pre-training is costly and possible only for a few large companies. Rather than trying to cover all kinds of test distributions during pre-training, we propose to add a third stage: editing the featurizer after fine-tuning. To this end, we interpolate the featurizer with auxiliary featurizers trained on auxiliary datasets. This merging via weight averaging edits the main featurizer by including the features mechanisms learned on the auxiliary datasets. Empirically, we show that this editing strategy improves the performance of existing state-of-the-art models on the DomainBed benchmark by adapting the featurizer to the test domain. We hope to encourage updatable approaches beyond the direct transfer learning strategy.

1 Introduction: the art of fine-tuning

Distribution shifts between train and test data can reduce the performance of deep neural networks [1, 2]. Thus, learning models that generalize well to unseen domains is critical as those models are now vastly deployed for real-world applications in many different contexts [3, 4]. The current paradigm for domain generalization is **transfer learning** [5, 6]: we pre-train a model on a large dataset and then fine-tune it on the available domains for the task.

The goal of the pre-training is to obtain the best possible featurizer. Though different learning objectives generalize differently [7, 8, 9, 10], recent works suggest that the main differences lie in the pre-training data distribution [11, 12, 13, 14] and the learnt features diversity [15, 16, 17]. To mitigate distribution shifts, this dataset should ideally contain diverse images with diverse concepts; then, it would transfer diverse features that can match domains [18], facilitate the optimization [19, 20], and overall generalize well on all kinds of distributions. Yet, under a limited pre-training budget [12], **the ideal pre-training is task-dependent**. Amortizing the computational cost is at the cornerstone of the "foundation models" paradigm [21]. Yet, only a few companies are capable of collecting the necessary huge-and-diverse data and deploying the hardwares. This is a major limitation to the community, as it comes with risks of centralization, monetization, privacy issues and lack of transparency [22]. To reduce the reliance on proprietary models, there is a need of alternative strategies involving individual stakeholders starting from open-source datasets such as ImageNet [23].

A promising lead is **inter-training**: recent works [24, 25, 26, 27, 28, 29] leveraged auxiliary datasets to enrich the featurizer. They fine-tune the model on a task, "intermediate" between the pre-training and the target tasks. We show the benefits but also the main limitation of inter-training: the auxiliary task should be adapted to the test domain, yet unavailable at initialization in domain generalization.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

In this paper, we propose a novel **editing** strategy to enrich the features; we edit the featurizer after fine-tuning, as illustrated in Figure 1. Our featurizer is interpolated towards featurizers trained on auxiliary datasets. Operating after training enables us to choose the adequate interpolating coefficients at test time, and thus to cheaply adapt the featurizer in function of the test domain. This is a radical departure from existing transfer learning strategies, that tried to anticipate all kind of distributions at initialization. Applied on the reference DomainBed benchmark [30] in Section 4, editing after training improves the performance of state-of-the-art domain generalization strategies.

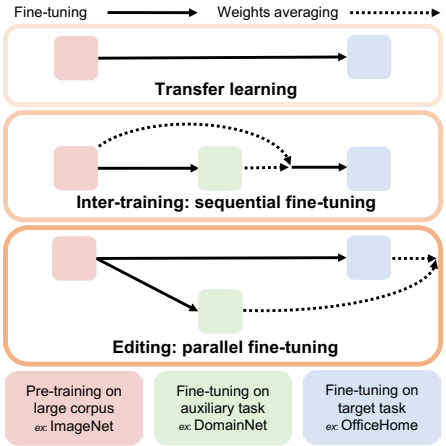

Figure 1: **Different learning strategies**.

Our main contributions are the following:

1. We highlight a key limitation of the inter-training strategy in Section 2.
2. We show that we can enrich the featurizer after fine-tuning and propose in Section 3 a three-stage strategy to improve domain generalization; pre-training on a large corpus, fine-tuning on the available training domains and then editing with auxiliary datasets.

Related work. Our strategy follows the "updatable machine learning" [31] literature, further discussed in Section 5. Notably, we are inspired by the recent success of weight averaging (WA) to merge models trained independently [32, 33], even when trained on different tasks [34, 35, 36]. Our work is also related to the editing literature [37, 38, 39] which tries to correct mistakes as they appear: none tackled domain generalization nor leveraged weight averaging.

2 Inter-training before fine-tuning

A promising strategy to enrich features. As illustrated in Figure 1, inter-training adds an intermediate learning task on an auxiliary dataset between pre-training and fine-tuning on the target task. The idea is to leverage similar datasets to enrich and specialize the features. Indeed, in the vicinity of the target task there are similar auxiliary tasks that would facilitate the learning of the adequate features. With ImageNet [23] as the pre-training dataset, we experiment inter-training on DomainNet [25] for domain generalization on OfficeHome [40] in Figure 2. The initialization is thus $(1-\lambda)\theta_{\text{ImageNet}} + \lambda\theta_{\text{DN}}$ where θ_{ImageNet} are the ImageNet weights and θ_{DN} are those weights specialized on DomainNet. We observe that full inter-training ($\lambda = 1$) improves the accuracy on the out-of-distribution (OOD) domain "Art" compared to the direct transfer learning from ImageNet ($\lambda = 0$).

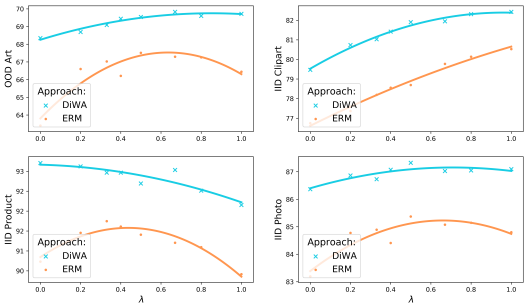

Figure 2: **Domain accuracy on OfficeHome after inter-training on DomainNet**. Our networks are initialized from $(1-\lambda)\theta_{\text{ImageNet}} + \lambda\theta_{\text{DN}}$ where θ_{ImageNet} are ResNet50 ImageNet weights and θ_{DN} those specialized on DomainNet. From each initialization, we train 20 networks on the "Clipart", "Product" and "Photo" domains, leaving apart the OOD domain "Art". Finally, we either average all weights (DiWA [33]) or select the best one based on the train-domain validation accuracy (ERM).

Inter-training requires a priori knowledge from the test domain. Yet, inter-training suffers from catastrophic forgetting [41] of the features previously learnt on ImageNet: learning new features does not fully preserve those from initialization. This explains why the inter-training on DomainNet is detrimental on the domain "Product": $\lambda \approx 0$ performs best for DiWA. Thus, a more "robust initialization" can usually be found for $0 < \lambda < 1$ in a similar fashion to "robust fine-tuning" [42]; by interpolating between the two extremes, we can find a better trade-off between pre-trained features and auxiliary features. Critically, the best value for λ is domain-dependant, and cannot be detected using the other domains; this finding contrasts with recent works [43] observing "a clear linear

relationship between IID accuracy and OOD accuracy". In conclusion, inter-training helps only when the auxiliary task is selected with some prior knowledge of the future test domains; DomainNet helps for OfficeHome and PACS [44] but will be detrimental for TerraIncognita [45] in Section 4.

3 Editing after fine-tuning

We argued in the previous section that we need some knowledge of the future test domains to best enrich the features with auxiliary tasks. This motivates a more flexible strategy that would be applied after fine-tuning, when some test samples could be available. To this end, we propose our editing strategy, illustrated in Figure 1; we leverage features from the auxiliary tasks not at initialization, but at inference time, without requiring a retraining step. In brief, we merge the featurizers fine-tuned on the target task with those fine-tuned on auxiliary tasks. To do so, we leverage weights averaging [32, 46], which enriches features (see Appendix B) and empirically succeeds to merge networks even when trained on different tasks [34, 35, 36] despite the non-linearities [47, 48]. Thus we predict using $(1-\lambda)\theta_{\text{target}} + \lambda\theta_{\text{aux}}$ as the featurizer; the classifier (on top of the featurizer) remains the one fine-tuned on the target task as the auxiliary tasks may not have the same classes.

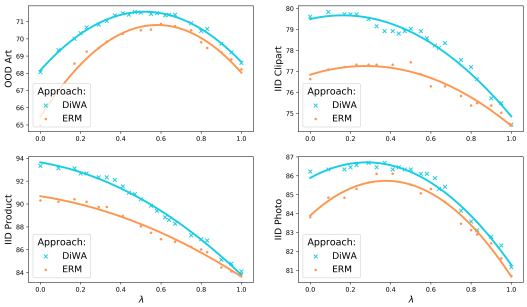

Figure 3: **Domain accuracy on OfficeHome after editing towards DomainNet**. Our networks are initialized from the ImageNet weights θ_{ImageNet}. We then train 20 networks on the "Clipart", "Product" and "Photo" domains, leaving apart the OOD domain "Art". To obtain θ_{HOME}, we either average all weights (DiWA [33]) or select the best one based on the train-domain validation accuracy (ERM). At test time, we use $(1-\lambda)\theta_{\text{HOME}} + \lambda\theta_{\text{DN}}$ where θ_{DN} are specialized on DomainNet.

We validate the benefits of editing for OfficeHome in Figure 3 and Appendix D, and on PACS in Appendix E. We note that editing consistently helps. We also remark similarities with the curve obtained via inter-training in Figure 2; the main advantage is that the editing operation is done after fine-tuning, and thus can be adapted cheaply to the test domain.

4 Experimental results on the DomainBed benchmark

Setup. We evaluate our strategy on the DomainBed [30] benchmark, which imposes the code, the training procedures and the ResNet50 [49] architecture. Each domain is successively considered as the test while others are for training: domains are split into 80% (used as training and evaluation) and 20% (used as validation). The featurizer is pre-trained on ImageNet, and the classifier is initialized with linear probing [50] to prevent features distortion: from this shared initialization, we fine-tune the whole network with 20 hyperparameter configurations sampled from Table 5. For ERM, θ_{target} is the one with highest validation accuracy. DiWA [33] uniformly averages the weights from all training runs, removing the need for model selection. We apply DiWA on DomainNet [25] to obtain θ_{aux} by averaging the 20 networks trained on DomainNet. At inference time, we use $(1-\lambda)\theta_{\text{target}} + \lambda\theta_{\text{aux}}$. λ is either arbitrarily set to 0.5, or optimized on the validation dataset of the test domain: in the latter case, we select the λ with the highest validation accuracy among 20 sampled from the uniform distribution $U(0,1)$. Thus, for fair comparison, scores are reported with DomainBed's test-domain model selection. This experimental setup is further described in Appendix C.

Results. In OfficeHome [40] in Table 1 and PACS [44] in Table 2, editing and inter-training consistently improve ERM and DiWA. The gains are larger for domains where DomainNet brings useful information, e.g., on the "Clipart" domain from OfficeHome, and smaller on others, e.g., on the "Product" domain. $\lambda = 0.5$ works well because DomainNet was selected with some human's prior as the auxiliary task for OfficeHome and PACS; yet, $\lambda = 0.5$ fails for TerraIncognita [45] in Table 3 because this dataset is not related to DomainNet's domains. In contrast, editing on TerraIncognita with oracle-domain validation for $\lambda \sim U(0,1)$ does not harm accuracies as $\lambda \approx 0$ are selected (see Appendix F). This test-time adaptation after fine-tuning is not possible with inter-training, which therefore performs badly. This highlights the main advantage of editing w.r.t. inter-training; when the auxiliary dataset is not relevant, the editing strategy can ignore the auxiliary weights at no cost.

Table 1: **Accuracy** (%, ↑) **on OfficeHome** (best in **bold** and second best underlined).

Algorithm	Transfer strategy	Art	Clipart	Product	Photo	Avg
ERM	Direct transfer learning	65.6	55.0	78.1	79.3	69.5
	Inter-training	67.9	66.6	79.2	81.3	73.8
	Editing: $\lambda = 0.5$	70.1	64.7	79.7	82.1	74.2
	Editing: λ best from $U(0,1)$	69.9	67.2	79.6	81.7	74.6
DiWA	Direct transfer learning	68.1	58.8	80.0	82.1	72.2
	Inter-training	69.4	**68.6**	**81.3**	82.1	75.4
	Editing: $\lambda = 0.5$	**71.5**	65.6	80.2	82.6	75.0
	Editing: λ best from $U(0,1)$	71.4	67.7	80.4	**82.9**	**75.6**

Table 2: **Accuracy** (%, ↑) **on PACS** (best in **bold** and second best underlined).

Algorithm	Transfer strategy	Art	Cartoon	Photo	Sketch	Avg
ERM	Direct transfer learning	90.3	82.4	98.8	83.0	88.6
	Inter-training	91.4	89.8	98.8	87.4	91.9
	Editing: $\lambda = 0.5$	92.5	85.3	**99.2**	87.2	91.0
	Editing: λ best from $U(0,1)$	92.6	85.3	**99.2**	87.2	91.1
DiWA	Direct transfer learning	90.4	82.7	98.7	82.5	88.6
	Inter-training	**93.5**	**90.2**	98.5	**88.2**	**92.6**
	Editing: $\lambda = 0.5$	91.7	84.8	99.1	86.1	90.4
	Editing: λ best from $U(0,1)$	91.2	84.0	99.1	86.2	90.1

Table 3: **Accuracy** (%, ↑) **on TerraIncognita** (best in **bold** and second best underlined).

Algorithm	Transfer strategy	L100	L38	L43	L46	Avg
ERM	Direct transfer learning	**58.7**	**51.2**	**60.8**	**42.0**	**53.2**
	Inter-training	57.9	47.2	56.6	39.9	50.4
	Editing: $\lambda = 0.5$	50.6	43.8	50.0	38.9	45.8
	Editing: λ best from $U(0,1)$	**58.7**	51.0	**60.8**	**42.0**	53.1
DiWA	Direct transfer learning	57.7	49.4	60.5	39.5	51.8
	Inter-training	56.7	45.2	59.1	37.1	49.5
	Editing: $\lambda = 0.5$	47.8	37.7	52.5	37.9	44.0
	Editing: λ best from $U(0,1)$	57.7	49.4	60.5	39.2	51.7

Limitations and future work. We will investigate the impact of other auxiliary tasks, e.g., iNaturalist [51] to improve performance on TerraIncognita. We will also report scores with standard deviation for different pre-training strategies [52] and architectures [53], and open-source our code. Finally, we will analyze the arithmetic properties of fine-tuning and weights averaging [54].

5 Updatable machine learning for domain generalization

The DomainBed benchmark [30] showed that no invariance-based approaches [55, 56, 57] outperformed ERM for **domain generalization**. Recently, **weight averaging** [46, 58] (WA) strategies did by averaging weights obtained along a training run [59, 60] or from multiple independent runs [32, 33] to increase diversity in predictions: similarly, editing with auxiliary tasks increases diversity in features. WA relies on the linear mode connectivity across networks [61, 62], which notably holds when networks are trained from a shared pre-trained initialization on a shared task [63]. Yet, recent findings suggest that WA may succeed even with different initializations [64, 65] — modulo a simple weights permutation — and on different tasks [34, 35, 36]. The latter works are the most similar to us. [34] motivates a Fisher-averaging in NLP; in contrast, we show that basic interpolation is efficient and discuss the selection of the interpolating coefficient. [35] proposed an "embarrassingly simple parralelization". [36] is specialized for zero-shot CLIP [52] models. Interestingly, our approach is also similar to "Robust fine tuning" [42]; while they rely on the initialization to generalize well, we learn appropriate auxiliary weights to edit the fine-tuned weights. These works complement each other and all highlight the potential of the **updatable machine learning** [31] paradigm.

Conclusion. We proposed a strategy for domain generalization: we edit the featurizer after fine-tuning by interpolating towards weights trained on auxiliary tasks. We validate this approach on the DomainBed benchmark. In contrast with centralized foundation models, we could leverage a repository of averageable specialized networks trained in a decentralized way, akin to a github repository for softwares [31]. We hope to pave the way towards deep models training in collaboration.

References

[1] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019. (p. 1)

[2] John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. The effect of natural distribution shift on question answering models. In *ICML*, 2020. (p. 1)

[3] John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLOS Medicine*, 2018. (p. 1)

[4] Emanuele Alberti, Antonio Tavera, Carlo Masone, and Barbara Caputo. Idda: a large-scale multi-domain dataset for autonomous driving. *RA-L*, 2020. (p. 1)

[5] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. (p. 1)

[6] Florian Wenzel, Andrea Dittadi, Peter Vincent Gehler, Carl-Johann Simon-Gabriel, Max Horn, Dominik Zietlow, David Kernert, Chris Russell, Thomas Brox, Bernt Schiele, Bernhard Schölkopf, and Francesco Locatello. Assaying out-of-distribution generalization in transfer learning. *arXiv preprint*, 2022. (p. 1)

[7] Mert Bulent Sariyildiz, Yannis Kalantidis, Diane Larlus, and Karteek Alahari. Concept generalization in visual representation learning. In *ICCV*, 2021. (p. 1)

[8] Simon Kornblith, Ting Chen, Honglak Lee, and Mohammad Norouzi. Why do better loss functions lead to less transferable features? *NeurIPS*, 2021. (p. 1)

[9] Donghyun Kim, Kaihong Wang, Stan Sclaroff, and Kate Saenko. A broad study of pre-training for domain generalization and adaptation. *arXiv preprint*, 2022. (p. 1)

[10] Yuge Shi, Imant Daunhawer, Julia E Vogt, Philip HS Torr, and Amartya Sanyal. How robust are pre-trained models to distribution shift? In *ICML SCIS Workshop*, 2022. (p. 1)

[11] Alex Fang, Gabriel Ilharco, Mitchell Wortsman, Yuhao Wan, Vaishaal Shankar, Achal Dave, and Ludwig Schmidt. Data determines distributional robustness in contrastive language image pre-training (CLIP). In *ICML*, 2022. (p. 1)

[12] Thao Nguyen, Gabriel Ilharco, Mitchell Wortsman, Sewoong Oh, and Ludwig Schmidt. Quality not quantity: On the interaction between dataset design and robustness of clip. *arXiv preprint*, 2022. (p. 1)

[13] Priya Goyal, Quentin Duval, Isaac Seessel, Mathilde Caron, Mannat Singh, Ishan Misra, Levent Sagun, Armand Joulin, and Piotr Bojanowski. Vision models are more robust and fair when pretrained on uncurated images without supervision. *arXiv preprint*, 2022. (p. 1)

[14] Dustin Tran, Jeremiah Liu, Michael W. Dusenberry, Du Phan, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, Neil Band, Tim G. J. Rudner, Karan Singhal, Zachary Nado, Joost van Amersfoort, Andreas Kirsch, Rodolphe Jenatton, Nithum Thain, Honglin Yuan, Kelly Buchanan, Kevin Murphy, D. Sculley, Yarin Gal, Zoubin Ghahramani, Jasper Snoek, and Balaji Lakshminarayanan. Plex: Towards reliability using pretrained large model extensions. *arXiv preprint*, 2022. (p. 1)

[15] Firas Laakom, Jenni Raitoharju, Alexandros Iosifidis, and Moncef Gabbouj. Within-layer diversity reduces generalization gap. *arXiv preprint*, 2021. (p. 1)

[16] Niv Nayman, Avram Golbert, Asaf Noy, Tan Ping, and Lihi Zelnik-Manor. Diverse imagenet models transfer better. *arXiv preprint*, 2022. (p. 1)

[17] Saachi Jain, Dimitris Tsipras, and Aleksander Madry. Combining diverse feature priors. In *ICML*, 2022. (p. 1)

[18] Yangjun Ruan, Yann Dubois, and Chris J. Maddison. Optimal representations for covariate shift. In *ICLR*, 2022. (p. 1)

[19] Jianyu Zhang, David Lopez-Paz, and Léon Bottou. Rich feature construction for the optimization-generalization dilemma. In *ICML*, 2022. (p. 1)

[20] Anonymous. Learning useful representations for shifting tasks and distributions. In *Submitted to ICLR*, 2023. under review. (p. 1)

[21] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint*, 2021. (p. 1)

[22] Rishi Bommasani and Percy Liang. Reflections on foundation models. https://hai.stanford.edu/news/reflections-foundation-models. Accessed: 2022-10-04. (p. 1)

[23] 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. *IJCV*, 2015. (pp. 1 and 2)

[24] Jason Phang, Thibault Févry, and Samuel R Bowman. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint*, 2018. (p. 1)

[25] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. (pp. 1, 2, 3, and 10)

[26] Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. Exploring and predicting transferability across NLP tasks. In *EMNLP*, 2020. (p. 1)

[27] Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel Bowman. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *ACL*, 2020. (p. 1)

[28] Sara Beery, Yang Liu, Dan Morris, Jim Piavis, Ashish Kapoor, Neel Joshi, Markus Meister, and Pietro Perona. Synthetic examples improve generalization for rare classes. In *ACV*, 2020. (p. 1)

[29] Gustav Mårtensson, Daniel Ferreira, Tobias Granberg, Lena Cavallin, Ketil Oppedal, Alessandro Padovani, Irena Rektorova, Laura Bonanni, Matteo Pardini, Milica G Kramberger, et al. The reliability of a deep learning model in clinical out-of-distribution mri data: a multicohort study. *MIA*, 2020. (p. 1)

[30] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *ICLR*, 2021. (pp. 2, 3, 4, and 9)

[31] Colin Raffel. A call to build models like we build open-source software. https://colinraffel.com/blog/a-call-to-build-models-like-we-build-open-source-software.html. Accessed: 2022-10-04. (pp. 2 and 4)

[32] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*, 2022. (pp. 2, 3, 4, and 9)

[33] Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. In *NeurIPS*, 2022. (pp. 2, 3, 4, 9, 10, 11, and 12)

[34] Michael Matena and Colin Raffel. Merging models with fisher-weighted averaging. In *NeurIPS*, 2022. (pp. 2, 3, and 4)

[35] Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models. *arXiv preprint*, 2022. (pp. 2, 3, and 4)

[36] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. *arXiv preprint*, 2022. (pp. 2, 3, and 4)

[37] Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitry Pyrkin, Sergei Popov, and Artem Babenko. Editable neural networks. In *ICLR*, 2020. (p. 2)

[38] Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. Editing a classifier by rewriting its prediction rules. *NeurIPS*, 34, 2021. (p. 2)

[39] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *ICLR*, 2022. (p. 2)

[40] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. (pp. 2, 3, and 9)

[41] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. (p. 2)

[42] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Hanna Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *CVPR*, 2022. (pp. 2 and 4)

[43] 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 *ICML*, 2021. (p. 2)

[44] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017. (pp. 3 and 9)

[45] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV*, 2018. (pp. 3 and 9)

[46] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *UAI*, 2018. (pp. 3, 4, and 9)

[47] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. (p. 3)

[48] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint*, 2018. (p. 3)

[49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. (pp. 3 and 10)

[50] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *ICLR*, 2022. (pp. 3 and 10)

[51] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *CVPR*, 2018. (p. 4)

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. (p. 4)

[53] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. (p. 4)

[54] Anonymous. Editing models with task arithmetic. In *Submitted to ICLR*, 2023. (p. 4)

[55] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint*, 2019. (p. 4)

[56] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *ICML*, 2021. (p. 4)

[57] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, 2016. (p. 4)

[58] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *ICML*, 2018. (p. 4)

[59] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. In *NeurIPS*, 2021. (pp. 4, 10, 11, 12, and 14)

[60] Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. Ensemble of averages: Improving model selection and boosting performance in domain generalization. *arXiv preprint*, 2021. (pp. 4 and 10)

[61] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *ICML*, 2020. (p. 4)

[62] Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *NeurIPS*, 2019. (p. 4)

[63] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In *NeurIPS*, 2020. (p. 4)

[64] Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *ICLR*, 2022. (p. 4)

[65] Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint*, 2022. (p. 4)

[66] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017. (p. 9)

[67] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. (p. 10)

[68] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. (p. 10)

[69] Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *arXiv preprint*, 2020. (pp. 10, 11, 12, and 14)

Appendices

This supplementary material is organized as follows:

- Appendix A describes the different learning strategies with pseudo-equations.
- Appendix B is a preliminary experiment suggesting that weight averaging enriches features.
- Appendix C further describes our experimental setup on DomainBed [30].
- Appendix D complements our experiments on OfficeHome [40].
- Appendix E complements our experiments on PACS [44].
- Appendix F complements our experiments on TerraIncognita [45].

A Learning strategies as pseudo-equations

In Figure 1, we illustrated the different learning strategies. In Equation (1), we now provide an analytical formulation of these strategies with pseudo-equations, where θ represents the featurizer's weights and $\mathcal{D}$ the datasets.

$$\begin{aligned}
&\theta = \text{Train}(\theta_{\text{pre-train}}, \mathcal{D}_{\text{target}}), && \text{[Direct transfer learning]} \\
&\theta = \text{Train}((1-\lambda)\theta_{\text{pre-train}} + \lambda \text{Train}(\theta_{\text{pre-train}}, \mathcal{D}_{\text{aux}}), \mathcal{D}_{\text{target}}), && \text{[Inter-training]} \\
&\theta = (1-\lambda)\text{Train}(\theta_{\text{pre-train}}, \mathcal{D}_{\text{target}}) + \lambda \text{Train}(\theta_{\text{pre-train}}, \mathcal{D}_{\text{aux}}). && \text{[Editing]}
\end{aligned} \tag{1}$$

B Similarity between weight averaging and representation averaging

Our editing strategy relies on the intuition that averaging weights enriches features: in this section, we present a preliminary experiment strengthening this intuition.

We consider two weights θ_1 and θ_2 obtained from two independent fine-tunings on OfficeHome from ImageNet weights. We are interested in their feature representation in the penultimate layer, just before the classifier, noted Φ_{θ_1} and Φ_{θ_2} when extracted over the validation dataset. The representation $(\Phi_{\theta_1} + \Phi_{\theta_2})/2$ averages the features extracted from both featurizers. We also consider the features extracted by their WA, i.e., $\Phi_{(\theta_1+\theta_2)/2}$. Table 4 shows the L_2 norm of the representation differences. The two closest representations are the averaged representation $(\Phi_{\theta_1} + \Phi_{\theta_2})/2$ ad the one from the weight averaged network $\Phi_{(\theta_1+\theta_2)/2}$. It suggests that WA successfully encodes the two representations to construct a richer one. In conclusion, while [32, 33, 46] argue that WA approximates predictions ensembling [66], this experiment suggests that WA also approximates features averaging.

Table 4: L_2 norm of the representation differences $||a - b||_2$.

	$\Phi_{(\theta_1+\theta_2)/2}$	$(\Phi_{\theta_1} + \Phi_{\theta_2})/2$	Φ_{θ_1}	Φ_{θ_2}
$\Phi_{(\theta_1+\theta_2)/2}$	0	2.4	5.7	5.5
$(\Phi_{\theta_1} + \Phi_{\theta_2})/2$		0	5.0	5.0
Φ_{θ_1}			0	10.0
Φ_{θ_2}				0

C DomainBed

We further detail our experiments on the DomainBed benchmark [30].

Training protocol. OfficeHome [40], PACS [44], and TerraIncognita [45] are divided into four domains. Each domain is successively considered as the test domain while other domains are used in training. We follow the training protocol from `https://github.com/alexrame/diwa`, which is itself adapted from `https://github.com/facebookresearch/DomainBed`. For each dataset and domain, we perform a random search of 20 trials on the mild hyperparameter distributions described

in Table 5. We use a ResNet50 [49] pre-trained on ImageNet, with a dropout layer before the newly added dense layer and fine-tuned with frozen batch normalization layers. The optimizer is Adam [67]. Our classifiers are initialized with linear probing [50]; we first learn only the classifier (with the featurizer frozen) with the default hyperparameters defined in Table 5; the classifier's weights are then used to initialize all subsequent runs. Following [33, 59, 60], for OfficeHome, PACS, and TerraIncognita, all runs are trained for 5k steps with validation accuracy computed every 100 steps; for DomainNet, all runs are trained for 15k steps with validation accuracy computed every 500 steps.

Table 5: Hyperparameters, their default values and distributions for random search.

Hyperparameter	**Default value**	**Random distribution**	
		Extreme (DomainBed)	Mild (Ours, DiWA and SWAD)
Learning rate	$5 \cdot 10^{-5}$	$10^{\mathcal{U}(-5,-3.5)}$	$[1, 3, 5] \cdot 10^{-5}$
Batch size	32	$2^{\mathcal{U}(3,5.5)}$	32
ResNet dropout	0	$[0, 0.1, 0.5]$	$[0, 0.1, 0.5]$
Weight decay	0	$10^{\mathcal{U}(-6,-2)}$	$[10^{-6}, 10^{-4}]$
Interpolating coefficient λ	0	$\mathcal{U}(0.1)$	

Baselines. ERM is the standard Empirical Risk Minimization. Mixup is the classical data augmentation [68] applied across domains (Interdomain Mixup [69]). SWAD [59] average weights along the trajectory of an ERM run. DiWA [33] uniformly averages the weights obtained from different ERM runs and removes the need of model selection.

Editing. A inference time, our featurizer is $(1-\lambda)\theta_{\text{target}} + \lambda\theta_{\text{DN}}$, where θ_{DN} is a featurizer specialized on DomainNet [25] and λ the interpolating hyperparameter. Specifically, θ_{DN} is obtained by applying DiWA's protocol on DomainBed for DomainNet, but with all domains available for training: thus θ_{DN} is the weights average of 20 networks fine-tuned on all six domains from DomainNet. Setting arbitrarily $\lambda = 0.5$ performs well on OfficeHome and PACS only because DomainNet was selected with some human prior information for these datasets; yet, it fails for TerraIncognita that contains images very different from DomainNet's.

Test-domain model selection. For more adaptive editing, needed notably for TerraIncognita, we can use a more principled approach based on test-domain model selection, i.e., when validation samples belong to the test domain. To do so, we sampled 20 values for λ from the uniform distribution $U(0, 1)$ and select the best one based on the test-domain validation accuracy. That's why, for fair comparison, the scores reported for baselines are those with test-domain model selection. In contrast with train-domain model selection, the test-domain model selection is arguably more realistic for real applications: indeed, after deployment of their models, the practitioners would validate that the performances are correct, and thus would label some samples. We will ablate the importance of the validation dataset's size in future revision.

D Editing on OfficeHome

D.1 Results on OfficeHome from DomainBed

Table 6: **Accuracy** (%, ↑) **on OfficeHome** (best in **bold** and second best underlined).

	Algorithm	Art	Clipart	Product	Photo	Avg
	ERM	61.7 ± 0.7	53.4 ± 0.3	74.1 ± 0.4	76.2 ± 0.6	66.4 ± 0.5
	Mixup [69]	63.5 ± 0.2	54.6 ± 0.4	76.0 ± 0.3	78.0 ± 0.7	68.0 ± 0.2
	SWAD [59]	66.1 ± 0.4	57.7 ± 0.4	78.4 ± 0.1	80.2 ± 0.2	70.6 ± 0.2
	DiWA [33]	68.4 ± 0.2	58.2 ± 0.5	80.0 ± 0.1	81.7 ± 0.3	72.1 ± 0.2
Our runs	ERM	65.6	55.0	78.1	79.3	69.5
	ERM + Inter-training	67.9	66.6	79.2	81.3	73.8
	ERM + Editing: $\lambda = 0.5$	70.1	64.7	79.7	82.1	74.2
	ERM + Editing: λ best from $U(0, 1)$	69.9	67.2	79.6	81.7	74.6
	DiWA	68.1	58.8	80.0	82.1	72.2
	DiWA + Inter-training	69.4	**68.6**	**81.3**	82.1	75.4
	DiWA + Editing: $\lambda = 0.5$	**71.5**	65.6	80.2	82.6	75.0
	DiWA + Editing: λ best from $U(0, 1)$	71.4	67.7	80.4	**82.9**	**75.6**

D.2 Editing curves w.r.t. λ

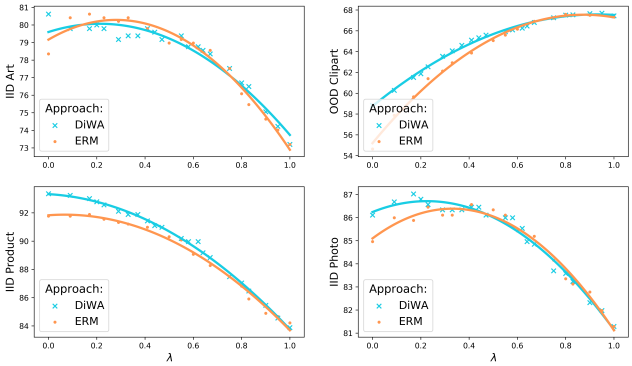

Figure 4: **Domain accuracy on OfficeHome with "Clipart" as OOD domain after editing towards DomainNet**. At test time, we use $(1 - \lambda)\theta_{\text{HOME}} + \lambda\theta_{\text{DN}}$.

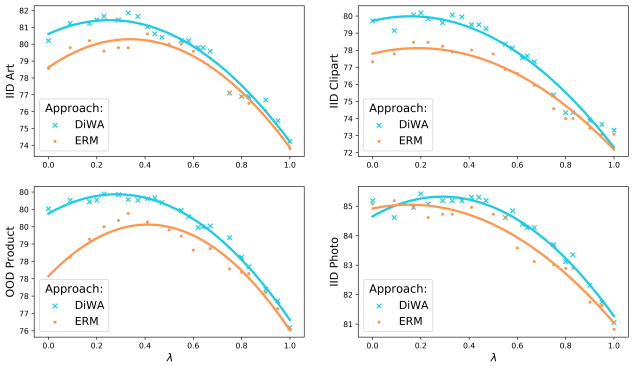

Figure 5: **Domain accuracy on OfficeHome with "Product" as OOD domain after editing towards DomainNet**. At test time, we use $(1 - \lambda)\theta_{\text{HOME}} + \lambda\theta_{\text{DN}}$.

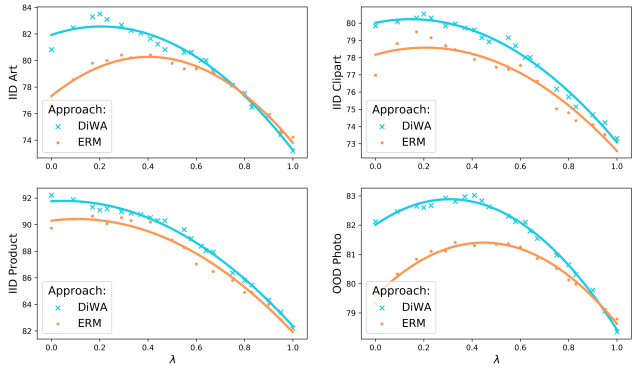

Figure 6: **Domain accuracy on OfficeHome with "Photo" as OOD domain after editing towards DomainNet**. At test time, we use $(1-\lambda)\theta_{\text{HOME}} + \lambda\theta_{\text{DN}}$.

E Editing on PACS

E.1 Results on PACS from DomainBed

Table 7: **Accuracy** ($\%, \uparrow$) **on PACS** (best in **bold** and second best underlined).

	Algorithm	Art	Cartoon	Photo	Sketch	Avg
	ERM	86.5 ± 1.0	81.3 ± 0.6	96.2 ± 0.3	82.7 ± 1.1	86.7 ± 0.3
	Mixup [69]	87.5 ± 0.4	81.6 ± 0.7	97.4 ± 0.2	80.8 ± 0.9	86.8 ± 0.3
	SWAD [59]	89.3 ± 0.5	83.4 ± 0.6	97.3 ± 0.3	82.5 ± 0.8	88.1 ± 0.1
	DiWA [33]	90.1 ± 0.2	82.8 ± 0.6	98.3 ± 0.1	83.3 ± 0.4	88.7 ± 0.2
Our runs	ERM	90.3	82.4	98.8	83.0	88.6
	ERM + Inter-training	91.4	89.8	98.8	87.4	91.9
	ERM + Editing: $\lambda = 0.5$	92.5	85.3	**99.2**	87.2	91.0
	ERM + Editing: λ best from $U(0,1)$	92.6	85.3	**99.2**	87.2	91.1
	DiWA	90.4	82.7	98.7	82.5	88.6
	DiWA + Inter-training	**93.5**	**90.2**	98.5	**88.2**	**92.6**
	DiWA + Editing: $\lambda = 0.5$	91.7	84.8	99.1	86.1	90.4
	DiWA + Editing: λ best from $U(0,1)$	91.2	84.0	99.1	86.2	90.1

E.2 Editing curves w.r.t. λ

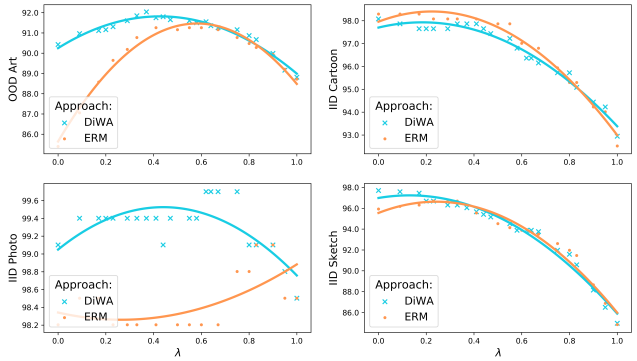

Figure 7: **Domain accuracy on PACS with "Art" as OOD domain after editing towards DomainNet**. At test time, we use $(1-\lambda)\theta_{\text{PACS}} + \lambda\theta_{\text{DN}}$.

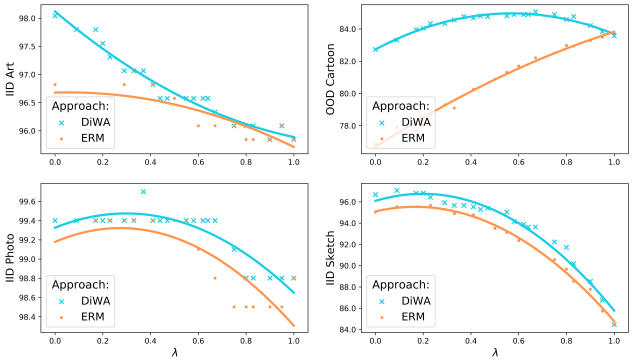

Figure 8: **Domain accuracy on PACS with "Cartoon" as OOD domain after editing towards DomainNet**. At test time, we use $(1-\lambda)\theta_{\text{PACS}} + \lambda\theta_{\text{DN}}$.

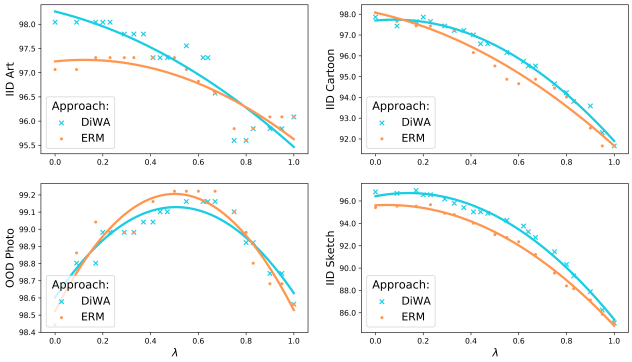

Figure 9: **Domain accuracy on PACS with "Photo" as OOD domain after editing towards DomainNet**. At test time, we use $(1-\lambda)\theta_{\text{PACS}} + \lambda\theta_{\text{DN}}$.

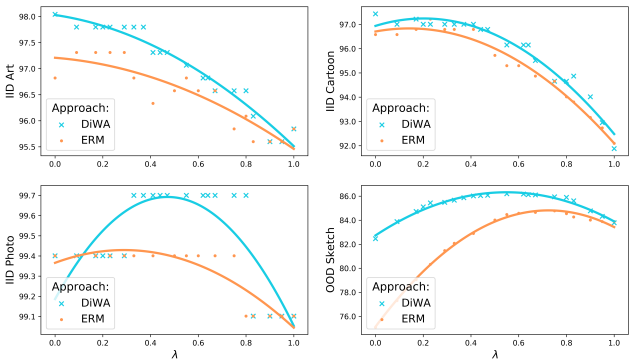

Figure 10: **Domain accuracy on PACS with "Sketch" as OOD domain after editing towards DomainNet**. At test time, we use $(1-\lambda)\theta_{\text{PACS}} + \lambda\theta_{\text{DN}}$.

F Editing on TerraIncognita

F.1 Results on TerraIncognita from DomainBed

Table 8: **Accuracy (%, ↑) on TerraIncognita** (best in **bold** and second best underlined).

	Algorithm	L100	L38	L43	L46	Avg
	ERM	59.4 ± 0.9	49.3 ± 0.6	60.1 ± 1.1	**43.2** ± 0.5	53.0 ± 0.3
	Mixup [69]	**67.6** ± 1.8	51.0 ± 1.3	59.0 ± 0.0	40.0 ± 1.1	**54.4** ± 0.3
	SWAD [59]	55.4 ± 0.0	44.9 ± 1.1	59.7 ± 0.4	39.9 ± 0.2	50.0 ± 0.3
	DiWA	56.0 ± 2.5	48.9 ± 0.8	58.4 ± 0.2	40.6 ± 0.8	51.0 ± 0.7
Our runs	ERM	58.7	**51.2**	**60.8**	42.0	53.2
Our runs	ERM + Inter-training	57.9	47.2	56.6	39.9	50.4
Our runs	ERM + Editing: $\lambda = 0.5$	50.6	43.8	50.0	38.9	45.8
Our runs	ERM + Editing: λ best from $U(0, 1)$	58.7	51.0	**60.8**	42.0	53.1
Our runs	DiWA	57.7	49.4	60.5	39.5	51.8
Our runs	DiWA + Inter-training	56.7	45.2	59.1	37.1	49.5
Our runs	DiWA + Editing: $\lambda = 0.5$	47.8	37.7	52.5	37.9	44.0
Our runs	DiWA + Editing: λ best from $U(0, 1)$	57.7	49.4	60.5	39.2	51.7

F.2 Editing curves w.r.t. λ

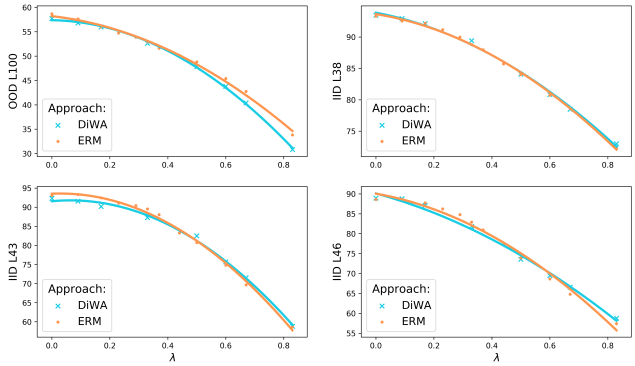

Figure 11: **Domain accuracy on TerraIncognita with "L100" as OOD domain after editing towards DomainNet**. At test time, we use $(1 - \lambda)\theta_{\text{TERRA}} + \lambda\theta_{\text{DN}}$.

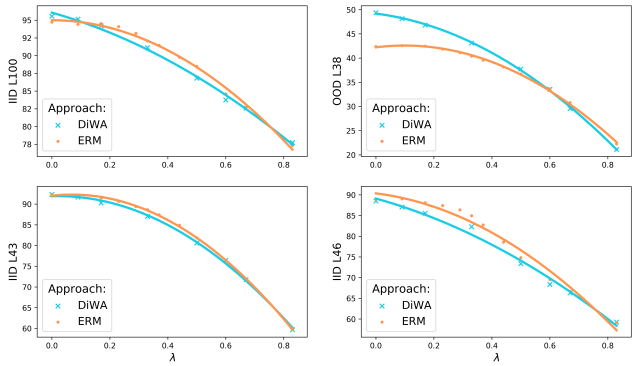

Figure 12: **Domain accuracy on TerraIncognita with "L38" as OOD domain after editing towards DomainNet**. At test time, we use $(1 - \lambda)\theta_{\text{TERRA}} + \lambda\theta_{\text{DN}}$.

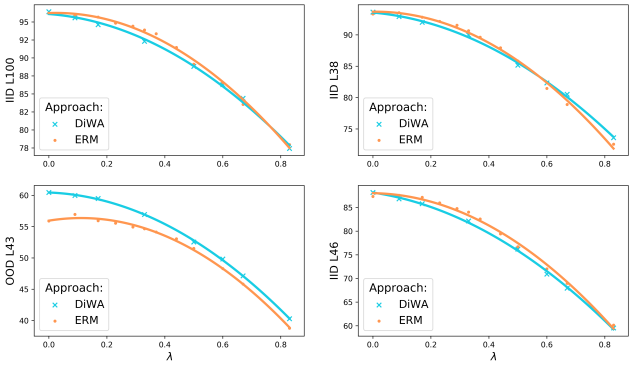

Figure 13: **Domain accuracy on TerraIncognita with "L43" as OOD domain after editing towards DomainNet**. At test time, we use $(1-\lambda)\theta_{\text{TERRA}} + \lambda\theta_{\text{DN}}$.

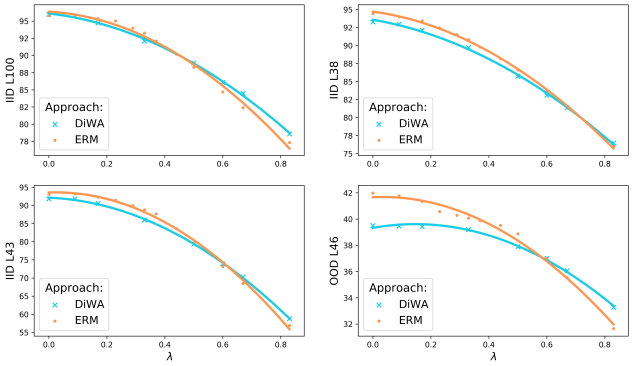

Figure 14: **Domain accuracy on TerraIncognita with "L46" as OOD domain after editing towards DomainNet**. At test time, we use $(1-\lambda)\theta_{\text{TERRA}} + \lambda\theta_{\text{DN}}$.