Pruning for Better Domain Generalizability

Xinglong Sun [1]

Abstract

In this paper, we investigate whether we could use pruning as a reliable method to boost the generalization ability of the model. We found that existing pruning method like L2 can already offer small improvement on the target domain performance. We further propose a novel pruning scoring method, called DSS, designed not to maintain the source accuracy of the compressed model as typical pruning work does, but to directly enhance the robustness and the generalization performance of the model. We conduct empirical experiments to validate our method and demonstrate that it can be even combined with state-of-the-art generalization work like MIRO(Cha et al., 2022) to further boost the performance. On MNIST to MNIST-M, we could improve the baseline performance by over 5 points simply by introducing 60% channel sparsity selected by DSS into the model. On the popular DomainBed benchmark and combining with MIRO, we can further boost the state-of-the-art performance by 1 point only by introducing 10% sparsity into the model. Code can be found at `https://github.com/AlexSunNik/pruning_for_domain_gen`.

1. Introduction

In recent years, we have seen rapid development in deep neural networks with various model architectures(He et al., 2016; Dosovitskiy et al., 2020) for a wide range of different tasks. However, the pretrained models could be poor at generalizing learned features or knowledge to new datasets or environments. Even a slight shift from the network's original training domain could significantly hurt its performance(Recht et al., 2019; Hendrycks & Dietterich, 2021; Yang et al., 2021), which suggests that the successes achieved by deep learning so far have been largely driven by supervised learning and large-scale labeled datasets(Deng et al., 2009). This domain shift issue has seriously impeded the development and practical deployment of deep neural models.

A straightforward approach to deal with this domain shift is to collect some data from the target domain to adapt the source-domain trained model, relying on the assumption that target data is accessible for model adaption. This *domain adaptation* approach(Lu et al., 2020; Saito et al., 2018; Ganin & Lempitsky, 2015; Long et al., 2015; Liu et al., 2020; Hoffman et al., 2018; Gong et al., 2012; Long et al., 2016; Balaji et al., 2019; Kang et al., 2019; Kulis et al., 2011; Gandelsman et al., 2022; Sun et al., 2020; Liu et al., 2021) is practical in many scenarios since we do not need extra labor and computation cost to label the collected target distribution data and simply leverage the unlabeled data for adaptation. However, it does assume that we have access to target domain data during training, which is infeasible and impractical in many cases.

A more strictly defined problem is *domain generalization*, which does not assume access to target sample features during training and strives to learn robust representations against distribution shifts from multiple source domains during training. We later perform evaluation of the trained model on an unseen test domain to measure the generalizability, transferability, and robustness of the model. While many existing works in domain generalization attempt to learn domain-invariant features (Arjovsky et al., 2019; Bui et al., 2021; Cha et al., 2021; Ganin & Lempitsky, 2015; Li et al., 2018a; Sun & Saenko, 2016), some recent works (Gulrajani & Lopez-Paz, 2021; Koh et al., 2021) also demonstrate decent accuracies without explicitly enforcing invariance. Recently, the state-of-the-art work MIRO (Cha et al., 2022) aims to learn similar features to 'oracle' representations, reformulating the domain generalization task by maximizing mutual information between oracle representations and model representations while performing training on source domains. It achieves state-of-art performance on the widely-used DomainBed (Gulrajani & Lopez-Paz, 2021) benchmark.

A seemingly completely irrelevant direction, pruning (LeCun et al., 1989; Hassibi et al., 1993; Han et al., 2015; Frankle & Carbin, 2019; Lee et al., 2019; Sun et al., 2022),

[1] Department of Computer Science, Stanford University. Correspondence to: Xinglong Sun <xs15@stanford.edu>.

Published at the ICML 2023 Workshop on Spurious Correlations, Invariance, and Stability. Honolulu, Hawaii, USA. Copyright 2023 by the author(s).

aims to compress the model by removing the least important channels scored by some saliency criterion. Since size of the model gets shrinked after pruning, it is also sometimes considered to reduce the overfitting of models and increase the domain generalization ability of the model from another perspective. In fact, some latest works (Li et al., 2022; Jin et al., 2022; Bartoldson et al., 2020) began to investigate deeper relationship between pruning and the generalizability and robustness of models.

In this paper, we made a further step, investigating whether we could use pruning as a reliable method to boost the generalization ability of the model. We aim to answer the following three questions:

1. Can we leverage existing popular and simple pruning metrics like L2(Li et al., 2017) to boost generalization accuracy by pruning unimportant channels?
2. Can we design a better pruning score taking the generalization ability of the model into consideration? More concretely, a score specifically designed to improve target domain accuracy instead of maintaining source domain accuracy as typical pruning.
3. Finally, can we combine it with modern state-of-the-art domain generalization algorithms like MIRO(Cha et al., 2022) as a simple plug-in component to further boost the accuracy?

We answered the above three questions with solid empirical studies ranging across three datasets and model architectures. To begin with, we study the first two questions extensively across many different pruning sparsity ratios on MNIST to MNIST-M, which is randomly colored MNIST. We found that the existing simple pruning method L2(Li et al., 2017) can offer a small improvement over the vanilla baseline(i.e. without using any domain generalization technique). Later, we solve the question (2) by designing a novel pruning method specifically targeting generalization accuracy. Given a convolutional neural networks(CNNs), we evaluate the activation map for samples from different domains at each layer for every channel and compute a *domain similarity score (DSS)* based on the distance of the activation maps. We then use *structural pruning* to prune the channels with the lowest DSS, followed by a finetuning session to recover accuracy. From empirical results, we observe an obvious improvement from the standard L2 pruning score. Notably, we can improve the baseline performance by more than 5 points by sparsifying 60% of the channels in the model, which may seem very surprising.

After validating the effectiveness of our proposed DSS score, we resolve the question (3) by combining our method with the state-of-the-art work MIRO (Cha et al., 2022). We conduct experiments on two datasets PACS and OfficeHome from the DomainBed (Gulrajani & Lopez-Paz, 2021) benchmark and observe a 1 point improvement of MIRO by introducing a 10% channel sparsity into the model, demonstrating the capability of our method to even improve the SOTA result.

2. Related Works

2.1. Domain Generalization

The problem of domain generalization was first introduced by (Blanchard et al., 2011) as a machine learning problem. As mentioned, the main difference of domain generalization from adaptation is that target-domain data is considered to be inaccessible during model training. Specifically in computer vision which usually involves dealing with several large-scale datasets simultaneously, (Torralba & Efros, 2011) found that dataset biases could lead to poor generalization performance by conducting an experiment with object recognition models on six benchmark datasets. Since then, many works aim to tackle domain generalization tasks from many perspectives. One major approach is to learn domain-invariant features by either minimizing between-domain feature divergences (Ganin & Lempitsky, 2015; Li et al., 2019; Matsuura & Harada, 2020; Sun & Saenko, 2016; Zhao et al., 2020), robust optimization (Arjovsky et al., 2019; Cha et al., 2021), or augmenting source domain examples (Bai et al., 2021; Carlucci et al., 2019). Inspired by these works, our pruning DSS score *also aims to remove features that are the most domain-sensitive*, or in other words, keep the most domain-similar features to be used for the later downstream tasks. Recently, the state-of-the-art work MIRO (Cha et al., 2022) reformulates the domain generalization task by maximizing mutual information between oracle representations and model representations while performing training on source domains. In our experiments, we are also going to demonstrate that our method can be used to further improve MIRO.

2.2. Pruning

Network pruning methods can be roughly categorized as *unstructured pruning* and *structured pruning*. Unstructured pruning methods (LeCun et al., 1989; Hassibi et al., 1993; Han et al., 2015; Frankle & Carbin, 2019; Lee et al., 2019) removes individual neurons of less importance without consideration for where they occur. On the other hand, structured pruning methods (Li et al., 2017; Liu et al., 2017; Molchanov et al., 2017) prune parameters under structure constraints, for example removing convolutional filters. Most pruning methods focus on designing an importance score to reflect parameters' importance to the final output. The popular channel pruning score L2 (Li et al., 2017) is evaluated as the Frobenious Norm of the convolution kernels. In this work, we leverage structural pruning as well

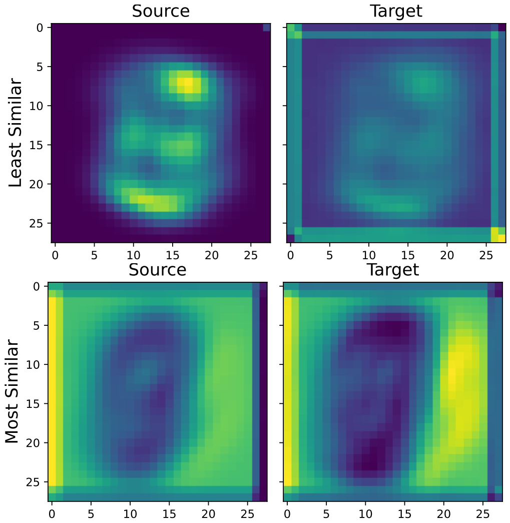

Figure 1. **MNIST to MNIST-M** Visualization of *averaged* feature maps that are least and most similar between source and target domains ranked by our DSS score. In the least similar case, we could see that the target feature map contains highlighted regions in the corner, which suggests potential *spurious features*.

because it allows us to filter and select features. Unlike all previous pruning works which aim to shrink the model size as much as possible while maintaining good source performance, our method focuses on enhancing the generalization ability of the model. We design a novel score *domain similarity score (DSS)* for this purpose, which measures the similarity of features between the source and target domain to keep robust features for estimating output.

3. Methodology

Here, we describe how to compute the proposed DSS score. Consider $\mathcal{X}$ as the input space and $\mathcal{Y}$ as the output space. We define a domain as a joint distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$. Moreover, we refer to P_X as the marginal or input distribution on X, and $P_{Y|X}$ as the posterior distribution of Y given X. Suppose we have two distinct domains, P^1_{XY} and P^2_{XY} ($P^2_{XY} \neq P^1_{XY}$).

Suppose we are dealing with image data and suppose $x^1 \sim P^1_X, x^1 \in \mathbb{R}^{c\times h\times w}$, and $x^2 \sim P^2_X, x^2 \in \mathbb{R}^{c\times h\times w}$, here, c, h, w are input channels(3 for RGB), height, and weight of our input data to the model. Below for convenience, when we do not distinguish the domain, we simply use x to refer to the input image. In representation learning, we usually build a feature extractor $T(.)$ constructing representation from raw input data x and outputting features maps or activation maps. The extracted features are then fed into a projector or linear classifier $f(.)$ to produce the final estimation, $i.e. f(T(x))$. Moreover, suppose $T(x) \in \mathbb{R}^{\hat{c}\times\hat{h}\times\hat{w}}$.

Given x^1 and x^2, we construct source and domain feature maps respectively as $T(x^1)$ and $T(x^2)$. The goal of our DSS score is to measure the similarity of $T(x^1)$ and $T(x^2)$. To this end, we simply leverage the score proposed in Deep-Face (Taigman et al., 2014) which computes the normalized inner product of the flattened features. We denote normalize and flatten operator as $\gamma(.)$ for simplicity. Therefore, at a convolutional layer, domain sensitivity score (DSS) $\mathcal{S}$ for each channel i out of $\hat{c}$ channels can be computed as:

$$\mathcal{S}_i = \langle \mathbb{E}[\gamma(T(x^1)_i)], \mathbb{E}[\gamma(T(x^2)_i)] \rangle \tag{1}$$

$$\mathcal{S} = [\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_{\hat{c}}] \tag{2}$$

In practice, we could leverage Monte Carlo Estimation for estimating the expected features from each distribution. For $\mathbb{E}[\gamma(T(x^1)_i)]$ for example, we can compute it as:

$$\mathbb{E}[\gamma(T(x^1)_i)] = \frac{1}{N} \sum_{x^1 \sim P^1_X}^{N} \gamma(T(x^1)_i), \tag{3}$$

which samples N times from the distribution P^1_X. A similar procedure can be done for P^2_{XY} as well. With the computed DSS score, we then prune the channels given by ArgBotK($\mathcal{S}, n$), which selects the bottom n channels out of $\hat{c}$ channels with the lowest DSS. We finish by performing a finetuning session on the kept channels to recover accuracy.

In Figure.1, we demonstrate the example feature maps on MNIST-M dataset with lowest(most sensitive) and highest (most similar) feature maps respectively. Interestingly, in the least similar case, we could see that the target feature map contains very highlighted regions in the corner of the image, which suggests potential *spurious features* since MNIST-like dataset mostly does not contain *'useful'* features there.

4. Empirical Results

As mentioned in Section1, we conduct experiments with the augmented MNIST dataset MNIST-M and two datasets PACS and OfficeHome from DomainBed (Gulrajani & Lopez-Paz, 2021). MNIST-M dataset contains digits from original MNIST dataset blended over patches randomly extracted from color photos of BSDS500. Since both datasets are relatively not large in size, in selecting N(number of times to estimate expected feature maps), we just leverage the entire dataset. Moreover, the methodology section only discusses the procedure on a single layer. With a multi-layered network like ResNet (He et al., 2016), we perform the described procedure at every layer independently.

4.1. Experiment Detail

All experiments are conducted with the PyTorch deep learning framework. Each experiment is performed on a NVIDIA Titan Xp GPU. In terms of evaluation, following standard

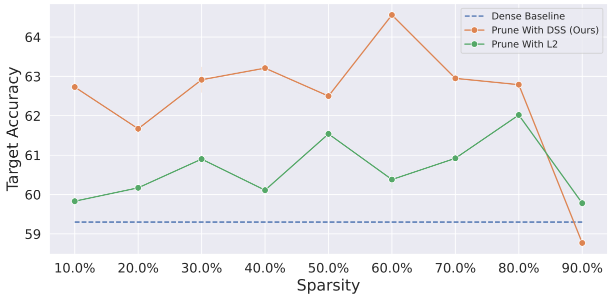

Figure 2. **MNIST to MNIST-M** Performance at different channel sparsity ratios. Our DSS scores reliably improve target accuracy and performs better than standard pruning scores like L2 (Li et al., 2017).

Method	Before Finetuning	After Finetuning
Baseline	59.30	
Rev.DSS	55.93	58.35
DSS	61.14	63.21

Table 1. **MNIST to MNIST-M** Ablation Results performed with pruning 40% channels with either our proposed DSS score or the reverse of our DSS score, i.e. we remove channels with highest DSS instead. The effectiveness of DSS is demonstrated in boosting the generalization performance of the model. Notably, with our proposed DSS score, even before fine-tuning immediately following pruning, we could already observe improvement.

practice, we train model on one or multiple source domains and evaluate it on the target domain.

4.2. MNIST-M

We conduct experiments on MNIST-M with a simple ConvNet which is also leveraged in works like (Ganin & Lempitsky, 2015). Here, we train on the vanilla MNIST data and evaluate on MNIST-M data for generalization performance and robustness. The two important comparisons here are (1) baseline, model only trained on the source MNIST and tested on MNIST-M and (2) baseline pruned with standard pruning metric L2 (Li et al., 2017). We prune at various channel sparsity ratios to better understand the behaviors. Results and comparisons are summarized in Figure2.

4.2.1. Optimization Detail

In terms of the optimization and hyperparameters of the baseline, we use the same settings as studies like (Ganin & Lempitsky, 2015). Concretely, the cross-entropy loss is leveraged as training loss for the classification task, and SGD optimizer is used for loss minimization. The learning rate is set as 0.01, and momentum is set as 0.9. Moreover, the batch size is set as 32, and the total training epochs is set as 100.

In terms of finetuning hyperparameters, we follow the same as the baseline. Instead, we only train an additional 20 epochs for finetuning.

4.2.2. Result Analysis

We can first observe that, standard and existing simple pruning method L2 (Li et al., 2017) can already offer a small improvement across various sparsity levels. For example, with 60% channel sparsity introduced into the model selected by L2, baseline performance can be boosted from 59.3 to around 60.5. This answers our question (1) raised in Sec.1 and confirms our expectation that general pruning can improve the generalization ability of the model.

Next, we can observe that our proposed DSS score further improves the results by a margin. Notably, we can improve the baseline performance by more than 5 points from 59.3 to 64.5 by sparsifying 60% of the channels in the model selected by DSS, which surpasses standard pruning score like L2. This answers our question (2) raised in Sec.1 and preliminarily demonstrates the effectiveness of our proposed method.

4.2.3. Ablation

Here, we also conduct a quick ablation study in verifying the effectiveness of our proposed method. Results are presented in Table 1. We compare with pruning with the reverse of our metric, which means that instead of removing the most sensitive features with the smallest DSS score, we remove the most similar features with the highest DSS score. Expectedly, we observe that the method degrades the performance of the baseline model. Surprisingly, we observe that, with DSS, even before fine-tuning, we can observe an improvement over the baseline from 59.30 to 61.14, further strengthening the effectiveness of our score.

4.3. DomainBed

As promised, we could also combine our proposed method with any state-of-the-art algorithm and further boost the performance. The experiment here is conducted on datasets PACS and OfficeHome from DomainBed (Gulrajani & Lopez-Paz, 2021). On PACS and OfficeHome, each contains four datasets, we follow the standard *leave-one* procedure which performs training and evaluation four times with each time training on three distributions and testing on the other. Final score is then averaged over these four runs. Moreover, our DSS score is also computed with an average of all of the three training domains. The important baseline here is thus naturally the model trained with the state-of-the-art algorithm MIRO (Cha et al., 2022).

Method	PACS	OfficeHome	Avg.
DANN (Ganin & Lempitsky, 2015)	83.6	65.9	74.8
CDANN (Li et al., 2018b)	82.6	65.8	74.2
IRM (Arjovsky et al., 2019)	83.5	64.3	73.9
GroupDRO (Sagawa et al., 2019)	84.4	66.0	75.2
ARM (Zhang et al., 2021)	85.1	64.8	75.0
ERM (Vapnik & Vapnik, 1998)	84.2	67.6	75.9
Mixup (Wang et al., 2020; Xu et al., 2020; Yan et al., 2020)	84.6	68.1	76.4
SelfReg (Kim et al., 2021)	85.6	67.9	76.8
MIRO (Cha et al., 2022)	85.4	70.5	78.0
MIRO + Ours 10%	$\mathbf{86.5}$	$\mathbf{71.4}$	$\mathbf{79.0}$

Table 2. **Domain Bed** Domain generalization results of applying our proposed method on state-of-the-art method MIRO. We reliably improve the performance of MIRO by shrinking the model size down by 10% removing channels selected by our DSS score.

4.3.1. Optimization Detail

We develop based on code provided by MIRO (Cha et al., 2022) and follow the same suggested optimization settings. Concretely, ResNet50 is selected as the base model architecture to perform the study. The cross-entropy loss is leveraged as training loss for the classification task, and the Adam optimizer is used for loss minimization. Default betas parameters of $0.9, 0.999$ are used for the Adam optimizer. The entire training procedure lasts for 5000 iterations. For PACS, the regularization lambda is set as 0.01 for MIRO training as described in the paper (Cha et al., 2022). The learning rate is set as $3e-5$ with no dropout and 0 weight decay. For OfficeHome, the regularization lambda is set as 0.1. The learning rate is set as $3e-5$ with 0.1 dropout and $1e-6$ weight decay.

For the finetuning following the pruning session, the learning rate is decayed by 10.

4.3.2. Result Analysis

Results are presented in Table.2. As observed in the table, we can further improve the state-of-the-art MIRO performance by introducing 10% channel sparsity selected by our method. Compared with MIRO (Cha et al., 2022), on PACS, we improve it from 85.4 to 86.5; on OfficeHome, we improve it from 70.5 to 71.4. On average, we improve MIRO from 78.0 to 79.0, a whole point improvement over the state-of-the-art only by introducing sparsity into the model. Given the amount of improvement of the research works on this benchmark as shown in Table 2, our improvement is quite decent, especially considering there is no cost for running DSS but to only inject channel sparsity into a pre-trained model. This answers our question (3) raised in Sec.1 and demonstrates the efficacy of the proposed score which can work in a model-agnostic way to improve potentially any domain generalization algorithm.

5. Discussion

In this paper, we made an initial investigation into the problem of leveraging pruning as a way to perform domain generalization and designed a simple and straightforward pruning score. A smarter and more adaptive approach is to incorporate trimming 'spurious' channels/features into the training objective itself and dynamically adjust model structure on-the-fly during training. We leave this for future study and hope this work, by looking at domain generalization from a novel pruning perspective, will inspire more following works.

6. Conclusion

In this paper, we study whether we could use pruning as a reliable method to improve the generalization performance of the model. We first found that existing pruning methods like L2 (Li et al., 2017) can already offer small improvement on the target domain performance. We then propose a novel pruning score DSS, designed not to maintain source accuracy of the pruned model, but to directly enhance the robustness and generalization performance of the model. We conduct empirical experiments to validate our method and demonstrate that it can be even combined with state-of-the-art generalization work like MIRO(Cha et al., 2022) to further boost the performance. We hope that our work offers a novel perspective on domain generalization, by reformulating it as a robust features selection or spurious features pruning task.

7. Acknowledgements

Thanks to Professor Tatsunori Hashimoto and his great course CS 329D at Stanford for inspiration to this paper.

References

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Bai, H., Sun, R., Hong, L., Zhou, F., Ye, N., Ye, H.-J., Chan, S.-H. G., and Li, Z. Decaug: Out-of-distribution generalization via decomposed feature representation and semantic augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6705–6713, 2021.

Balaji, Y., Chellappa, R., and Feizi, S. Normalized wasserstein for mixture distributions with applications in adversarial learning and domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6500–6508, 2019.

Bartoldson, B., Morcos, A., Barbu, A., and Erlebacher, G. The generalization-stability tradeoff in neural network pruning. *Advances in Neural Information Processing Systems*, 33:20852–20864, 2020.

Blanchard, G., Lee, G., and Scott, C. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24, 2011.

Bui, M.-H., Tran, T., Tran, A., and Phung, D. Exploiting domain-specific features to enhance domain generalization. *Advances in Neural Information Processing Systems*, 34:21189–21201, 2021.

Carlucci, F. M., D'Innocente, A., Bucci, S., Caputo, B., and Tommasi, T. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2229–2238, 2019.

Cha, J., Chun, S., Lee, K., Cho, H.-C., Park, S., Lee, Y., and Park, S. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021.

Cha, J., Lee, K., Park, S., and Chun, S. Domain generalization by mutual-information regularization with pre-trained models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, pp. 440–457. Springer, 2022.

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

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Gandelsman, Y., Sun, Y., Chen, X., and Efros, A. Test-time training with masked autoencoders. *Advances in Neural Information Processing Systems*, 35:29374–29385, 2022.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Gong, B., Shi, Y., Sha, F., and Grauman, K. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 2066–2073. IEEE, 2012.

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

Han, S., Pool, J., Tran, J., and Dally, W. J. Learning both weights and connections for efficient neural networks. *Neural Information Processing Systems (NIPS)*, 2015. URL http://dblp.uni-trier.de/db/journals/corr/corr1506.html#HanPTD15.

Hassibi, B., Stork, D. G., and Wolff, G. J. Optimal brain surgeon and general network pruning. In *IEEE international conference on neural networks*, pp. 293–299. IEEE, 1993.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2021.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., and Darrell, T. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pp. 1989–1998. Pmlr, 2018.

Jin, T., Carbin, M., Roy, D. M., Frankle, J., and Dziugaite, G. K. Pruning's effect on generalization through the lens of training and regularization. *arXiv preprint arXiv:2210.13738*, 2022.

Kang, G., Jiang, L., Yang, Y., and Hauptmann, A. G. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4893–4902, 2019.

Kim, D., Yoo, Y., Park, S., Kim, J., and Lee, J. Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9619–9628, 2021.

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

Kulis, B., Saenko, K., and Darrell, T. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR 2011*, pp. 1785–1792. IEEE, 2011.

LeCun, Y., Denker, J. S., Solla, S. A., Howard, R. E., and Jackel, L. D. Optimal brain damage. In *Neural Information Processing Systems(NIPs)*, volume 2, pp. 598–605. Citeseer, 1989.

Lee, N., Ajanthan, T., and Torr, P. H. S. Snip: single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations(ICLR)-Poster*, 2019. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2019.html#LeeAT19.

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

Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.-Z., and Hospedales, T. M. Episodic training for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1446–1455, 2019.

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient convnets. In *International Conference on Learning Representations(ICLR)-Poster*, 2017. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2017.html#0022KDSG17.

Li, Y., Gong, M., Tian, X., Liu, T., and Tao, D. Domain generalization via conditional invariant representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018b.

Li, Z., Chen, T., Li, L., Li, B., and Wang, Z. Can pruning improve certified robustness of neural networks? *arXiv preprint arXiv:2206.07311*, 2022.

Liu, Y., Kothari, P., Van Delft, B., Bellot-Gurlet, B., Mordan, T., and Alahi, A. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34:21808–21820, 2021.

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, pp. 2736–2744, 2017.

Liu, Z., Miao, Z., Pan, X., Zhan, X., Lin, D., Yu, S. X., and Gong, B. Open compound domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12406–12415, 2020.

Long, M., Cao, Y., Wang, J., and Jordan, M. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Long, M., Zhu, H., Wang, J., and Jordan, M. I. Unsupervised domain adaptation with residual transfer networks. *Advances in neural information processing systems*, 29, 2016.

Lu, Z., Yang, Y., Zhu, X., Liu, C., Song, Y.-Z., and Xiang, T. Stochastic classifiers for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9111–9120, 2020.

Matsuura, T. and Harada, T. Domain generalization using a mixture of multiple latent domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11749–11756, 2020.

Molchanov, P., Tyree, S., Karras, T., Aila, T., and Kautz, J. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations(ICLR)-Poster*, 2017. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2017.html#MolchanovTKAK17.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

Saito, K., Watanabe, K., Ushiku, Y., and Harada, T. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3723–3732, 2018.

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pp. 443–450. Springer, 2016.

Sun, X., Hassani, A., Wang, Z., Huang, G., and Shi, H. Disparse: Disentangled sparsification for multitask model compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12382–12392, 2022.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., and Hardt, M. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708, 2014.

Torralba, A. and Efros, A. A. Unbiased look at dataset bias. In *CVPR 2011*, pp. 1521–1528. IEEE, 2011.

Vapnik, V. and Vapnik, V. Statistical learning theory wiley. *New York*, 1(624):2, 1998.

Wang, Y., Li, H., and Kot, A. C. Heterogeneous domain generalization via domain mixup. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3622–3626. IEEE, 2020.

Xu, M., Zhang, J., Ni, B., Li, T., Wang, C., Tian, Q., and Zhang, W. Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 6502–6509, 2020.

Yan, S., Song, H., Li, N., Zou, L., and Ren, L. Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*, 2020.

Yang, J., Zhou, K., Li, Y., and Liu, Z. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

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

Zhao, S., Gong, M., Liu, T., Fu, H., and Tao, D. Domain generalization via entropy regularization. *Advances in Neural Information Processing Systems*, 33:16096–16107, 2020.

67.199+ 56.930+ 77.872+ 82.014