CADG: A Model Based on Cross Attention for Domain Generalization

Cheng Dai[1], Fan Li [1*], Xiyao Li[2]
Kuaishou Technology
{daicheng, lifan07, lixiyao}@kuaishou.com

Donglin Xie [3]
Sichuan University
2020223045208@stu.scu.edu.cn

Abstract

In Domain Generalization (DG) tasks, models are trained by using only training data from the source domains to achieve generalization on an unseen target domain, this will suffer from the distribution shift problem. So it's important to learn a classifier to focus on the common representation which can be used to classify on multi-domains, so that this classifier can achieve a high performance on an unseen target domain as well. With the success of cross attention in various cross-modal tasks, we find that cross attention is a powerful mechanism to align the features come from different distributions. So we design a model named CADG (cross attention for domain generalization), wherein cross attention plays a important role, to address distribution shift problem. Such design makes the classifier can be adopted on multi-domains, so the classifier will generalize well on an unseen domain. Experiments show that our proposed method achieves state-of-the-art performance on a variety of domain generalization benchmarks compared with other single model and can even achieve a better performance than some ensemble-based methods.

***K*eywords** Cross attention · Domain generalization · Transformer

1 Introduction

Currently, deep learning models have achieved remarkable performance in a wide range of aspects in the independent and identically distributed (i.i.d) setting. But in a real-word application, distribution shift caused by physical or psychological factors can always be found, which breaks the i.i.d assumption and make the models suffer from poor generalization. So more and more studies has focus to address this problem and domain generalization[1](DG) is one of the most popular and promising studies.

A data distribution can be seen as a domain and domain generalization (DG) aims to address distribution shift by learning a classifier that can generalize well on the test data sampled from an unseen distribution, after training on more than one training domains. In order to improve the generalization on the test domain in DG setting, various methods have been proposed. Some recent works demonstrate that using empirical risk minimization[2] (ERM) along with proper model selection (i.e. early stopping using validation set) under a fair evaluation protocol, called DomainBed [3], can already make the classifier to achieve a competitive performance against most previous domain generalization methods which focus on learning domain-invariant feature representations.

With the rising of DomainBed, more and more domain generalization methods focus on using some learning algorithms and model selection to achieve a better performance and one of the most popular and useful methods is ensemble learning[4] [5]. This make the less seeking on the model design for addressing the distribution shift problem and will not be helpful to touch the boundary of the generalization with a single model. Besides, a ensemble method will cause more resource cost in application phrase and more work in update phrase, so training a end-to-end model to address the distribution shift problem will be more applicable. In this case, we come back to research a better way to align the distributions of source and target domains.

Inspired by some other works involve the feature alignment such as cross-modal retrieval[6] and multi-modal representation[7], and with the success of the transformer used in CV aspect, we find that cross-attention is a powerful mechanism to align the feature come from different distributions. So we design a model named CADG (cross attention for domain generalization), wherein cross-attention play a important role, to address distribution shift problem. in our method, data come from different training domains with the same label will be organized as a data pair and we will use cross attention to extract the stable representation which can be used to classify from this pair.

Our contributions can be summarized as follow: 1. We introduce a new feature alignment way with cross attention to extract the stable representation which can be used to classify for domain generalization. As the model is based on transformer, so various performance improvement methods can be introduced in our method in future work. 2. Our proposed method achieves state-of-the-art performance on a variety of domain generalization benchmarks compared with other single models and can even outperform some ensemble-based methods.

2 Related work

2.1 Domain generalization

As [8] summarized, most existing domain generalization methods can be divided into following categories. (1) Data manipulation. This idea is first proposed in [9], which aims to create diverse training data to assist learning general representations on unseen target domain. Along this line, [10] and [11] improve the generalization of the models through domain randomization and some other works such as [12] use self-supervised learning to strengthen the generalization capability of the models. (2) Representation learning. This category of methods aims to find the invariant representation across various domains and one of the most popular techniques is domain alignment, which usually use some regularizations to align feature across domains explicitly such as maximum mean discrepancy [13] (MMD), second order correlation [14] and Wasserstein distance [15]. Our method belongs to representation learning but is a implicit way via cross-attention. (3) Learning strategy: This category of methods focuses on exploiting the general learning strategy to promote the generalization capability. Ensemble learning is one of the most available methods such as [16][17], but as we discussed before, these methods are not very friendly in a real-word application.

2.2 Transformer

Transformer[18] is first proposed to model sequential data in NLP tasks and has been proved excellent performance. In recent years, with the effectiveness of transformer has been showed in computer-vision tasks[19]. Through feeding transformer with sequences of image patches, such a VIT[20] and many other ViT variants[21][22], more and more studies have focused on using transformer based models to address CV tasks, cross-modal retrieval tasks and multi-modal representation tasks. Attention mechanism is the core of transformer and a extension of it named cross-attention has been proved very powerful at distilling noise and feature alignment in several works[7][23] for multi-modal based networks. This inspires us than if we regard the data from different modals as a special distribution, the situation that we should align features from different distributions in multi-modal tasks will be very similar with domain generalization tasks. So in this paper, we apply cross-attention to align feature across multi-domains to address the distribution shift problem.

3 The Proposed Method

We first introduce our proposed co-attention module and analyze its role in eliminating domain differences. Then the two-way center-aware training method is presented in Section 3.2 . With the constructed pair as sample input, our CDAG is proposed in Section 3.3, which consists of four weight-sharing transformers.

3.1 The Transformer Based Vision Cross Attention

In many computer vision tasks, vision transformer (ViT [20]) has achieved comparable or even superior performance. The self-attention([18]) module of ViT is one of the most important structure. ViT reshapes an image $I \in I^{H \times W \times C}$ into a sequence of flattened 2D patches $x \in R^{N \times (P^2 \cdot C)}$, where (H, W) is the height and width of the origin image, C is the channel numbers, P is the resolution of each image patch. Therefore, we can get a total of $N = H \cdot W/P^2$ patches. In self-attention, the image patches are first mapped into three vectors by projected, which are queries $Q \in R^{N \times d_q}$, keys $K \in R^{N \times d_k}$, and value $V \in R^{N \times d_v}$. The output of self-attention is obtained by weighted sum of the value, where the weight for each value is computed by the query and key, through a compatibility function. The self-attention module takes N patches as input and aims to mine the relationship between the patches of the input image. Its calculation process is as follows:

$$SelfAttn(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

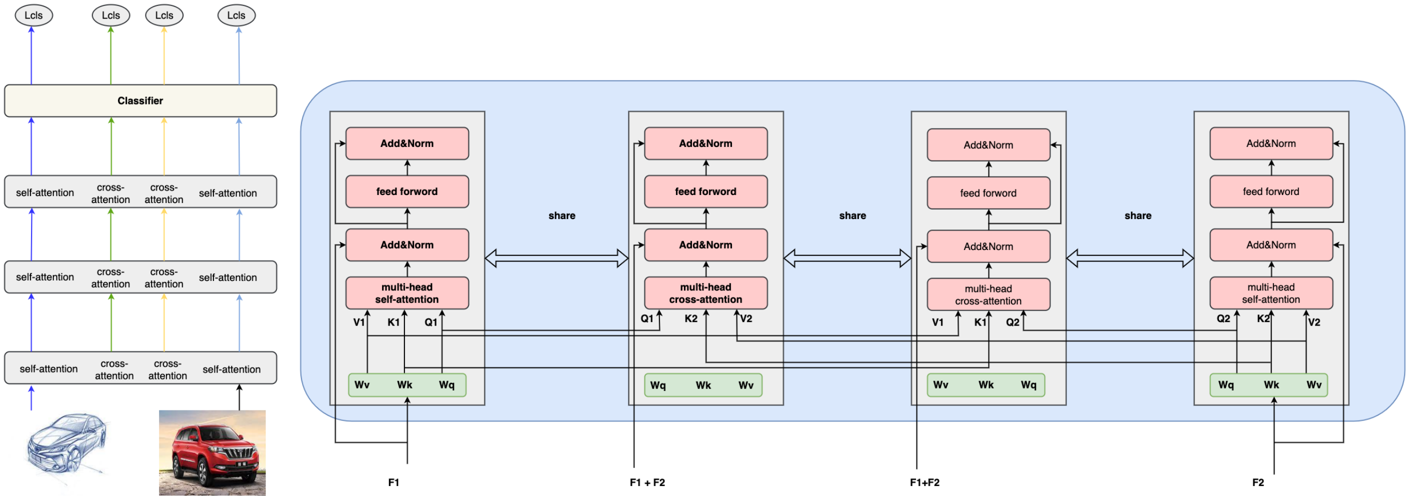

Figure 1: domain-generation transformer layer

The difference between self-attention and cross-attention if that the input of cross-attention is an images pair, such as I_1, I_2. The query of cross-attention is from I_1, while the key/value is from $I2$. Its calculation process is:

$$CrossAttn(Q_1, K_2, V_2) = softmax(\frac{Q_1 K_2^T}{\sqrt{d_k}})V_2$$

The M patches from I_1 consist of the queries $Q_1 \in R^{M \times d_k}$,while the $K_2 \in R^{N \times d_k}$ and $V_2 \in R^{N \times d_v}$ are keys and values from N patches of image I_2. Cross-attention has M outputs, each output is a weighted sum of V_t, and the weights are calculated by a compatibility function from the queries of I_1 and the keys in I_2. The result of cross-attention is that the path of I_2 will have a greater weight if it's more similar to query of I_1.

The input of cross-attention is an image pair from two different domains. By means of cross-attention, the alignment of different domains is realized in the latent space, and the difference between domains is eliminated. Specifically, for two images from different domains, split them into patches and perform cross-attention. Similar patches from different sources have greater attention weights than dissimilar patches. In this way, similar attributes between different domain data are preserved, and dissimilar ones are eliminated, so as to eliminate the differences between domains and extract domain invariant features.

3.2 Pair-wised Training Process

In general domain generalization research, data from multiple domains are mixed together, the model inputs a picture, and the model is monocentric. In this paper, we propose a pair-wised training method. First, we sample a category from the set of categories in the sample, then sample two domains from multiple training domains, and sample images from the corresponding categories of these two domains to construct a picture pair. Each image is fed into the model for self-attention, and then the image pair is fed into the model for cross-attention. The detailed algorithm flow is described in algorithm 1.

3.3 CADG: Cross Attention for Domain Generalization

The framework of the proposed method is shown in Figure 1. Our model consists of four weight-sharing transformer blocks. There are four data flows and constraints for the weight-sharing branches. It consists of 4-way weight-sharing transformer modules, two of which are self-attention modules named as self-branch1 and self-branch2. The others are cross-attention modules named as cross-branch1 and cross-branch2. The input to the framework is two images with the same label from two different domains. As shown in Fig1, the two images are input to two self-attention branches for learning domain-specific features. The outputs of the two branches are respectively connected to the softmax classifier for supervised classification training.

The inputs of the cross-attention come from the outputs of the two self-attention modules. For the N-th layer, the queries of the cross-branch1 module come from the queries of the N-th layer's self-branch1, and the keys and values

Algorithm 1 Two-Way Center-Aware Training

```
Initialization:$D = D_1, D_2, D_3, ...D_m$ are m domains
Initialization:$(x_i^j, y_i^j)$ is the j-th sample of domain $D_i$
Initialization:$C = C_1, C_2, ..., C_k$ are k classes, $epoch$ is the number of epochs, $S$ is the steps for an epoch.
Initialization:$model$, $e \leftarrow 0$, $s \leftarrow 0$
while $e < epoch$ do
    while $s < S$ do
        sample one class $C_r$ from C
        sample two domains $D_p, D_q$ from D
        sample a data $(x_p^i, y_p^i)$ from $D_p$ and another data $(x_q^j, y_q^j)$ from $D_q$, where $y_p^i = C_r$ and $y_q^j = C_r$
        $loss = model(x_p^i, x_q^j)$
        $backward(loss)$
        $s \leftarrow s + 1$
    end while
    $e \leftarrow e + 1$
end while
```

come from the corresponding self-branch2. The inputs of cross-branch2 is just the opposite of cross-branch1, which its queries come from the queries of the N-th layer's self-branch2, while the keys and values come from the corresponding self-branch1. The outputs of the cross-attention modules are added to the outputs of the (N-1)-th layer.

Each of the four branches will access a softmax classifier with shared parameters and calculate the cross entropy loss. The four losses are weighted to obtain the final classification loss.

$$L_{cls} = \lambda_1 loss_1 + \lambda_2 loss_2 + \lambda_3 loss_{cross1} + \lambda_4 loss_{cross2}$$

where $loss_1$ and $loss_2$ are the loss of the two self-attention branches, $loss_{cross1}$ and $loss_{cross2}$ are the cross-attention branch's losses.

4 Experiment

4.1 Datasets and Implementation

The proposed method is verified on 5 real-world benchmark datasets including PACS (4 domains, 9,991 samples, 7 classes) [24], VLCS (4 domains, 10,729 samples, 5 classes) [25], OfficeHome (4 domains, 15,588 samples, 65 classes) [26], TerraIncognita (4 domains, 24,778 samples, 10 classes) [27], and DomainNet (6 domains, 586,575 samples, 345 classes) [28].

We follow DomainBed's [3] training and validation protocols for fair comparisons. For model selection, we use the training domain validation set protocol. Specifically, the data of a certain domain is selected as the target domain, the rest are the source domain, and 20% of the source domain is divided as the validation set. We run training and validation 3 times, using different random seeds, with different splits of the training-validation set. The out-of-domain test performance averaged over all domains will be reported for each dataset. Except for DomainNet[28] which is 15000 steps, we use the standard 5000 iterations for other datasets, and reduce unnecessary computation by early-stop based on the validation set accuracy.

This section presents experimental results on the DomainBed suite. The input image size in our experiments is 224×224. We user the DeiT-base [29] which is pretrained on ImageNet1K as our backbone. We use the Stochastic Gradient Descent algorithm with the momentum of 0.9 and weight decay ratio 1e-4 to optimize the training process. The learning rate is set to 3e-3 for all datasets. The batch size is set to 64.

Firstly, our algorithm is compared with a non-ensemble algorithm, and the results are shown in Table 1. It can be found that our proposed method, without any tricks, outperforms all non-ensemble algorithms and reaches the sota level. In all experiments, our proposed method has significant improvement over non-ensemble methods: +3.1pp in PACS, +0.9pp in VLCS, +11.3pp in OfficeHome, +4.9pp in TerraIncognita and +7.1pp in DomainNet. On all datasets, our method has an overall average improvement of 6.6 pp compared to the non-ensemble algorithm.

We also compare our algorithm with the ensemble algorithm which are pretrained on ImageNet, and the results are shown in Table 2. It can be found that without any ensemble, our algorithm can surpass all ensemble algorithms on the list and become state-of-the-art with 1.1pp improvement. Comparing the best results of the ensemble algorithm, our algorithm outperforms on the two datasets: +0.5pp on PACS, +2.4pp on DomainNet.

Algorithm	PACS	VLCS	OfficeHome	TerraIncognita	DomainNet	avg
DANN (JMLR'16) [30]	84.6±1.1	78.7±0.3	65.4±0.6	48.4±0.5	38.4±0.0	63.1
CORAL (ECCV'16)[31]	86.0±0.2	77.7±0.5	68.6±0.4	46.4±0.8	41.8±0.2	64.1
MLDG (AAAI'18) [32]	84.8±0.6	77.1±0.4	68.2±0.1	46.1±0.8	41.8±0.4	63.6
MMD (CVPR'18) [33]	85.0±0.2	76.7±0.9	67.7±0.1	49.3±1.4	39.4±0.8	63.6
C-DANN (ECCV'18) [34]	82.8 ±0.5	78.2±0.4	65.6±0.5	47.6±0.8	38.9±0.1	62.6
ERM (ICLR'21) [3]	85.7±0.5	77.4±0.3	67.5±0.5	47.2±0.4	41.2±0.2	63.8
Fish (ICLR'22)[35]	85.5±0.3	77.8±0.3	68.6±0.4	45.1±1.3	42.7±0.2	63.9
CADG(ours)	**89.1 ±0.3**	**79.6 ±0.2**	**79.9 ±0.0**	**54.2 ±0.5**	**49.8 ±0.1**	**70.5**

Table 1: Non-ensemble algorithms

Algorithm	PACS	VLCS	OfficeHome	TerraIncognita	DomainNet	avg
SWAD (NIPS'21) [16]	88.1±0.1	79.1±0.1	70.6±0.2	50.0±0.3	46.5±0.1	66.9
EoA (arxiv) [17]	88.6	79.1	72.5	52.3	47.4	68.0
random ensemble	58.1±0.13	58.5±1.26	59.6±0.38	31.5±0.4	15.8±1.4	44.5
SEDGE[36]	84.1±0.45	**79.8±0.12**	**79.9±0.12**	**56.8±0.21**	46.3±0.39	69.4
CADG(ours)	**89.1 ±0.3**	79.6 ±0.2	79.9 ±0.0	54.2 ±0.5	**49.8 ±0.1**	**70.5**

Table 2: ensemble algorithms which pretrained on ImageNet

Although model pretrained on ImageNet is in common use, Kumar [37] finds that model pretrained on ImageNet may not be good for all datasets such as DomainNet. By using model that pretrained on the CLIP dataset [38], our model further improves the average performance by 2.2 pp .The details can be found in table 3.

Algorithm	PACS	VLCS	OfficeHome	TerraIncognita	DomainNet	avg
EoA$^+$ (arxiv) [17]	93.2	80.4	80.2	55.2	54.6	72.7
random ensemble	59.5±0.5	61.1±0.12	59.5±0.07	30.8±0.37	18.7±0.62	46.0
SEDGE$^+$ [36]	**96.1±0.04**	82.2±0.03	**80.7±0.21**	**56.8±0.29**	**54.7±0.1**	**74.1**
CADG$^+$ (ours)	94.6 ±0.42	**82.2 ±1.1**	79.5 ±0.5	55.7 ±0.2	51.6 ±0.0	72.7

Table 3: ensemble algorithms which pretrained on Clip

5 Conclusions

In this paper, we propose a single-model domain generalization algorithm based on cross-attention transformer, which can achieve feature alignment and extract invariant features. It does not require the use of ensemble methods, that greatly facilitates the application of such algorithms in practical scenarios. Based on this paper, subsequent work related to transformers can also be introduced into the task of domain generalization. Experiments on five benchmark datasets show that our proposed method achieves state-of-the-art performance compared with other single model and can even achieve a better performance than some ensemble-based methods.

References

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

[2] N. V. Vladimir and V. Vapnik, "Statistical learning theory," *Xu JH and Zhang XG. translation. Beijing: Publishing House of Electronics Industry, 2O04*, 1998.

[3] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," *arXiv preprint arXiv:2007.01434*, 2020.

[4] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[5] T. Zhou, S. Wang, and J. A. Bilmes, "Diverse ensemble evolution: Curriculum data-model marriage," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[6] G. Geigle, J. Pfeiffer, N. Reimers, I. Vulić, and I. Gurevych, "Retrieve fast, rerank smart: Cooperative and joint approaches for improved cross-modal retrieval," *arXiv preprint arXiv:2103.11920*, 2021.

[7] R. Hu and A. Singh, "Unit: Multimodal multitask learning with a unified transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1439–1449.

[8] J. Wang, C. Lan, C. Liu, Y. Ouyang, W. Zeng, and T. Qin, "Generalizing to unseen domains: A survey on domain generalization," *arXiv preprint arXiv:2103.03097*, 2021.

[9] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[10] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3803–3810.

[11] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 969–977.

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

[13] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE transactions on neural networks*, vol. 22, no. 2, pp. 199–210, 2010.

[14] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[15] F. Zhou, Z. Jiang, C. Shui, B. Wang, and B. Chaib-draa, "Domain generalization with optimal transport and metric learning," *arXiv preprint arXiv:2007.10573*, 2020.

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

[17] D. Arpit, H. Wang, Y. Zhou, and C. Xiong, "Ensemble of averages: Improving model selection and boosting performance in domain generalization," *arXiv preprint arXiv:2110.10832*, 2021.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[19] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on visual transformer," *arXiv e-prints*, pp. arXiv–2012, 2020.

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

[21] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 558–567.

[22] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[23] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov, "Multimodal transformer for unaligned multimodal language sequences," in *Proceedings of the conference. Association for Computational Linguistics. Meeting*, vol. 2019. NIH Public Access, 2019, p. 6558.

[24] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5542–5550.

[25] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR 2011*. IEEE, 2011, pp. 1521–1528.

[26] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5018–5027.

[27] S. Beery, G. Van Horn, and P. Perona, "Recognition in terra incognita," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 456–473.

[28] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1406–1415.

[29] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.

[30] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[31] X. Peng and K. Saenko, "Synthetic to real adaptation with generative correlation alignment networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 1982–1991.

[32] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[33] H. Li, S. J. Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5400–5409.

[34] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao, "Deep domain generalization via conditional invariant adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 624–639.

[35] Y. Shi, J. Seely, P. H. Torr, N. Siddharth, A. Hannun, N. Usunier, and G. Synnaeve, "Gradient matching for domain generalization," *arXiv preprint arXiv:2104.09937*, 2021.

[36] Z. Li, K. Ren, X. Jiang, B. Li, H. Zhang, and D. Li, "Domain generalization using pretrained models without fine-tuning," *arXiv preprint arXiv:2203.04600*, 2022.

[37] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang, "Fine-tuning can distort pretrained features and underperform out-of-distribution," *arXiv preprint arXiv:2202.10054*, 2022.

[38] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.