Leveraging Generative Foundation Models for Domain Generalization

Sobhan Hemati [1] Mahdi Beitollahi [1] Amir Hossein Estiri [1] Bassel Al Omari [1] Xi Chen [1] Guojun Zhang [1]

Abstract

There has been a huge effort to tackle the Domain Generalization (DG) problem with a focus on developing new loss functions. Inspired by the capabilities of the diffusion models, we pose a pivotal question: Can diffusion models function as data augmentation tools to address DG from a data-centric perspective, rather than relying on the loss functions? We show that trivial cross domain data augmentation (CDGA) along with the vanilla ERM using readily available diffusion models outperforms state-of-the-art (SOTA) DG algorithms. To justify the success of CDGA, we experimentally show that CDGA reduces the distribution shift between domains which is the main reason behind the lack of out-of-distribution (OOD) generalization of ERM under domain shift. These results advocate for further investigation into the potential of SOTA generative models for tackling the representation learning problem.

1. Introduction

Out-of-distribution (OOD) generalization is a pivotal ability for deep learning models in real-world scenarios. The prevalent setting for investigating OOD generalization is termed *domain generalization* (DG) Blanchard et al. (2011), involving multiple source domains to generalize to an unseen target domain. In DG problems, there is a shift between the training domains and the target domain which makes the models trained using Empirical Risk Minimization (ERM) (Vapnik, 1999) struggle to maintain their performance in the target domain. To enhance OOD generalization of ERM within the DG framework, researchers have proposed innovative loss functions based on different forms of invariant representation learning on feature level (Sun & Saenko, 2016; Ganin et al., 2016; Li et al., 2018; Tzeng et al., 2014), classifier head (Arjovsky et al., 2019), loss (Krueger et al., 2021) and gradient/Hessian (Parascandolo et al., 2020; Shahtalebi et al., 2021; Koyama & Yamaguchi, 2020; Shi et al., 2021; Hemati et al., 2023). Similar to our work, there is another line of work that explores data augmentation-based algorithms for DG (Gulrajani & Lopez-Paz, 2020; Somavarapu et al., 2020; Zhou et al., 2021; Carlucci et al., 2019; Ilse et al., 2021; Zhang et al., 2017). Nevertheless, none of these approaches consistently outperform others across all datasets, as illustrated by DomainBed benchmark (Gulrajani & Lopez-Paz, 2020). This observation suggests that a singular regularizer capable of capturing all invariances might not exist. Given the diverse shifts present in each dataset, encompassing correlation shift, diversity shift, label shift, etc. it is highly possible that a rigid, data-independent regularizer may not be able to mitigate different types of spurious correlations and shifts. Additionally, the incorporation of sub-optimal regularizers can impose excessive risk (Sener & Koltun, 2022), additional hyperparameters, and computational load to ERM.

Recent advances in diffusion-based generative models (Ho et al., 2020; Song et al., 2020; Rombach et al., 2022; Zhang & Agrawala, 2023) demonstrate their capability to achieve SOTA image quality. Recently, it has been shown that synthetic images generated by diffusion models can boost representation learning performance. Tian et al. (2023) showed in the self-supervised learning, synthetic images generated by stable diffusion models can enhance SimCLR (Chen et al., 2020). Inspired by these advancements in generative foundation models, rather than relying solely on traditional loss functions, we attempt to address the DG problem from a data-centric standpoint. Specifically, the capability of Denoising Diffusion Models (Ho et al., 2020; Song et al., 2020; Rombach et al., 2022) in generating high-fidelity synthetic images offers an innovative approach for advanced data augmentation, to enhance OOD generalization. To examine this hypothesis, we employ a straightforward *Cross Domain Generative Augmentation* (CDGA) method. In CDGA, synthetic images are generated conditioned on images or text descriptions from all possible combinations of the training domain pairs using a pre-trained latent diffusion model (LDM) (Rombach et al., 2022). We show that applying vanilla ERM along with generated and real images outperforms the previous state-of-the-art algorithms *across all datasets* in the DomainBed benchmark. Our empirical investigations show that generated synthetic images miti-

[1] Noah Ark Lab. Correspondence to: Sobhan Hemati <sobhan.hemati@huawei.com>.

Published at ICML 2024 Workshop on Foundation Models in the Wild. Copyright 2024 by the author(s).

gate the domain shift across domains while preserving the semantic information inherent to each class. From a theoretical standpoint, CDGA along with ERM is equivalent to replacing pointwise kernel estimates in ERM with new density estimates in the proximity of *domain pairs*. This modification to ERM reduces the inherent data estimation error in the presence of domain shift, subsequently enhancing its out-of-distribution (OOD) performance. To the best of our knowledge, we are the first to utilize latent diffusion models as a data-centric approach for DG.

2. Cross Domain Generative Augmentation

In this section, we provide a detailed description of CDGA. CDGA utilizes LDM to perform a transformation denoted by $\mathcal{M}(\cdot)$. This transformation takes two arguments as inputs: a data point in one domain and a guidance attribute in another domain from the same class. Formally,

$$\widetilde{x}_k^{i,j} = \mathcal{M}(x_k^i, \texttt{guide}^j), \tag{1}$$

where $\widetilde{x}_k^{i,j}$ is a synthetic image transformed from domains i and j, generated from the k-th sample in S_i. The attribute $\texttt{guide}^j$ serves as guidance towards another domain, S_j, within the same class.

In CDGA, each data point in domain S_i undergoes transformation to all n domains, including its own domain. This augmentation increases the number of samples for domain S_i from $|S_i|$ to $(b \times n + 1) \times |S_i|$, where n is the number of training domains, $|S_i|$ is the number of data points in S_i, and b is the generation batch size. Furthermore, we introduce CDGA*, where we assume access to a guidance attribute of the target domain. In this scenario, the size of domain S_i increases from $|S_i|$ to $(b \times (n+1) + 1) \times |S_i|$. the workflow is illustrated in Figure 1.

CDGA with Prompt Guidance (CDGA-PG): In CDGA-PG, given the k-th image in S_i, i.e., x_i^k, the guidance attribute $\texttt{guide}^j$ is a domain description text prompt that represents the same class in S_j. Having the image and the prompt guidance, we use the LDM to generate b synthetic images which we expect to interpolate domains i and j for the same class. For each image in S_i, we perform these image translations for all the training domains $j, \forall j \in \{1, ..., n\}$. We also consider the scenario where we can utilize the target domain description, i.e., $\texttt{guide}^{\mathcal{T}}$ as the guidance.

CDGA with Image Guidance (CDGA-IG): For scenarios where a text prompt description of domains is not available, CDGA-IG is used where the guidance is an image from S_j instead of a text description. More precisely, in CDGA-IG we attempt to mix two images from two different domains which is also known as the image mixer in the literature.

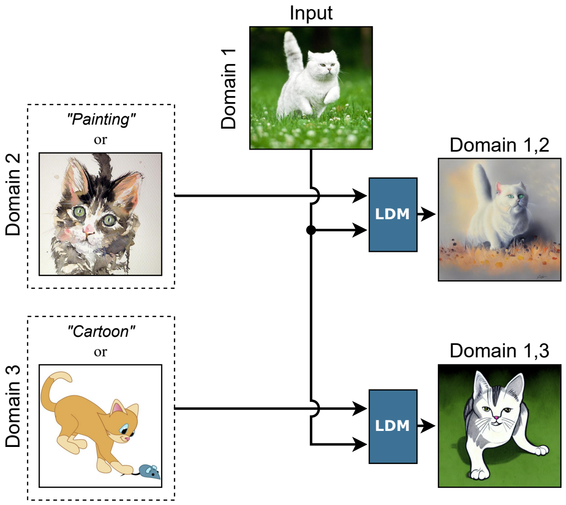


Figure 1: Illustration of CDGA. For each input image of a domain, we generate a new image using the image or the description of another domain. The generated image is an interpolation between two domains.

3. CDGA Outperforms SOTA

In this section, we compare CDGA + ERM with SOTA DG training methods, demonstrating its superior performance. We assess CDGA and CDGA* on for datasets, namely VLCS (Fang et al., 2013), PACS (Li et al., 2017), OfficeHome (Venkateswara et al., 2017), and DomainNet (Peng et al., 2019), using the DomainBed benchmark (Gulrajani & Lopez-Paz, 2020). This benchmark has gained popularity as a fair and standard evaluation platform for domain generalization algorithms. The evaluation process involves comparing DG algorithms across 20 hyperparameter choices and 3 trials, utilizing three distinct model selection techniques. To demonstrate CDGA's effectiveness, we present its evaluation results using the DomainBed benchmark in Tables 1-4. The tables follow a format of presenting the **first** and second results. For brevity, we report only the top five performing algorithms for each model selection, with full results available in the Appendix D. Examining Tables 1-4, CDGA* consistently achieves SOTA performance across all datasets and model selection techniques. Specifically, we applied prompt guidance for PACS, OfficeHome, and DomainNet, while using image guidance (i.e., image mixer) for VLCS. The code implementation for deploying CDGA-generated data within the DomainBed scheme is detailed in Appendix F.

4. CDGA Reduces Domain Shift

In this section, we empirically validate that CDGA reduces domain shift. To validate the efficacy of CDGA in mitigating domain shift, we employ five domain shift quantification

Table 1: DomainBed benchmark for **training-domain validation set** model selection method.

Algorithm	**PACS**	**OfficeHome**	**DomainNet**	**Avg**
ERM	85.5 ± 0.2	66.5 ± 0.3	40.9 ± 1.8	64.3
CORAL	86.2 ± 0.3	68.7 ± 0.3	41.5 ± 0.1	65.5
SagNet	86.3 ± 0.2	68.1 ± 0.1	40.3 ± 0.1	64.9
Fish	85.5 ± 0.3	68.6 ± 0.4	42.7 ± 0.2	65.6
Fishr	85.5 ± 0.4	67.8 ± 0.1	41.7 ± 0.0	65.0
HGP	84.7 ± 0.0	68.2 ± 0.0	41.1 ± 0.0	64.7
ERM + CDGA-PG	88.5 ± 0.5	68.2 ± 0.6	43.7 ±0.1	66.6
ERM + CDGA-PG*	**89.5** ± 0.3	**70.8** ± 0.6	**44.8** ±0.0	**68.4**

Table 2: DomainBed benchmark for **leave-one-domain-out cross-validation** model selection.

Algorithm	**PACS**	**OfficeHome**	**DomainNet**	**Avg**
ERM	83.0 ± 0.7	65.7 ± 0.5	40.6 ± 0.2	63.1
CORAL	82.6 ± 0.5	68.5 ± 0.2	41.1 ± 0.1	64.1
SagNet	82.3 ± 0.1	67.6 ± 0.3	40.2 ± 0.2	63.4
MLDG	82.9 ± 1.7	66.1 ± 0.5	41.0 ± 0.2	63.3
HGP	82.2 ± 0.0	67.5 ± 0.0	41.1 ± 0.0	63.6
Hutchinson	84.8 ± 0.0	68.5 ± 0.0	41.4 ± 0.0	64.9
ERM + CDGA-PG	86.8 ± 0.4	68.7 ± 0.4	43.6 ±0.1	66.2
ERM + CDGA-PG*	**88.4** ± 0.5	**70.2** ± 0.4	**44.8** ±0.0	**67.8**

techniques from the literature on the PACS dataset. Specifically, we utilize t-SNE visualization of feature embeddings, near-duplicate analysis (Oquab et al., 2023), and diversity shift metrics (Ye et al., 2022) to quantify the shift between domains.

4.1. Domain shift Visualization

To visualize domain shifts in CDGA-based data for the class "dog" across all domains (P, A, C, and S), we generate synthetic images for A → A, A → P, A → C, and A → S. Subsequently, we utilize the pretrained CLIP ViT-B/32 image encoder (Radford et al., 2021) to extract features from both real and synthetic images. These features are then projected onto a two-dimensional space using t-SNE and presented in Figure 2. Notably, the cross-domain synthetic images effectively interpolate between different domains, addressing the desired distribution shift. In Figure 2, examining domains A (in red) and S (in pink) reveals a significant distribution shift in their two-dimensional representations, despite all images belonging to the dog class. However, A → S synthetic images seamlessly bridge the gap between A and S representations. Refer to Figure 9 in the Appendix C for t-SNE plots of other classes.

4.2. Diversity Shift

Ye et al. (2022) proposed a numerical method to measure diversity shift which is equivalent to total variation (Zhang et al., 2021) to quantify domain shift. Diversity shift is usually due to the novel domain-specific features in the data. We employ the proposed algorithm by Ye et al. (2022) to quantify and compare diversity shift between training domains and the target domain in a leave-one domain out scheme for PACS real data, CDGA-PACA, and CDGA*-PACS datasets. Figure 5 (left) shows both CDGA and CDGA* reduce the diversity shift between training domains and the target domain.

Table 3: DomainBed benchmark **test-domain validation set (oracle)**model selection method.

Algorithm	**PACS**	**OfficeHome**	**DomainNet**	**Avg**
ERM	86.7 ± 0.3	66.4 ± 0.5	41.3 ± 0.1	64.8
Mixup	86.8 ± 0.3	68.0 ± 0.2	39.6 ± 0.1	64.8
MLDG	86.8 ± 0.4	66.6 ± 0.3	41.6 ± 0.1	65.0
CORAL	87.1 ± 0.5	68.4 ± 0.2	41.8 ± 0.1	65.8
SagNet	86.4 ± 0.4	67.5 ± 0.2	40.8 ± 0.2	64.9
Fish	85.8 ± 0.6	66.0 ± 2.9	43.4 ± 0.3	65.1
Fishr	86.9 ± 0.2	68.2 ± 0.2	41.8 ± 0.2	65.6
Hutchinson	86.3 ± 0.0	68.4 ± 0.0	41.9 ± 0.0	65.5
ERM + CDGA-PG	89.6 ± 0.3	68.8 ± 0.3	44.4 ±0.1	67.2
ERM + CDGA-PG*	**90.4** ± 0.3	**70.2** ± 0.2	**44.8** ±0.0	**68.5**

Table 4: DomainBed benchmark on **VLCS** dataset.

Method	Training domain	Leave-one -domain-out	Oracle
ERM	77.5 ± 0.4	77.2 ± 0.4	77.6 ± 0.3
CORAL	78.8 ± 0.6	78.7 ± 0.4	77.7 ± 0.2
SagNet	77.8 ± 0.5	77.5 ± 0.3	77.6 ± 0.1
Fishr	77.8 ± 0.1	78.2 ± 0.0	78.2 ± 0.2
HGP	77.6 ± 0.0	76.7 ± 0.0	77.3 ± 0.0
Hutchinson	76.8 ± 0.0	**79.3** ± 0.0	77.9 ± 0.0
ERM + CDGA-IG	**78.9** ± 0.3	77.9 ± 0.5	**79.5** ± 0.1

4.3. Near-duplicate Analysis

We employ near-duplicate image detection on images generated using CDGA to quantify the similarity between the generated and original images in each domain. Following the self-supervised image retrieval technique outlined in (Oquab et al., 2023), we utilize the pretrained CLIP ViT-B/32 image encoder (Radford et al., 2021) to extract embeddings and calculate cosine similarity between original and generated images. For each original image, if at least one image in a generated domain exhibits a cosine similarity above 0.95, we categorize the original domain as having a near-duplicate. Figure 3 provides a summarized view of this experiment for the case of generated images from domain C, while the complete results are available in Figure 7 in the Appendix B. In Figure 3, we report, for each original domain, the percentage of near-duplicates relative to the original domain size. Clearly, generating synthetic images

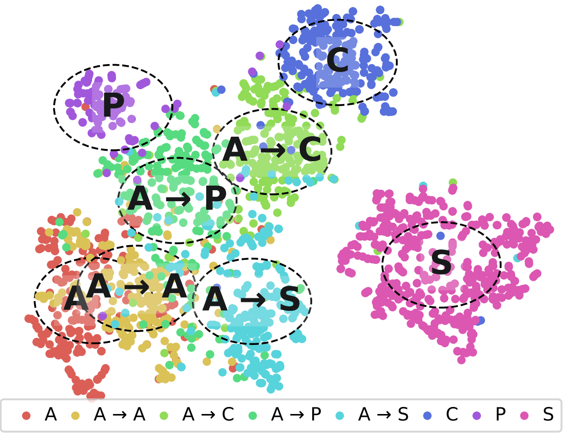
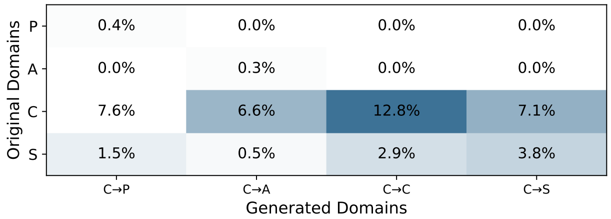


Figure 2: The t-SNE plot of features extracted from the original PACS dataset and generated images by CDGA from A domain. This figure shows that CDGA can fill the gap between the original domains. Check Section 4 for details.



	C→P	C→A	C→C	C→S
P	0.4%	0.0%	0.0%	0.0%
A	0.0%	0.3%	0.0%	0.0%
C	7.6%	6.6%	12.8%	7.1%
S	1.5%	0.5%	2.9%	3.8%

Figure 3: Heat map of the percentage of near-duplicates of each original domain in the generated domains. This table shows that using target-domain description results in more near-duplicate images.

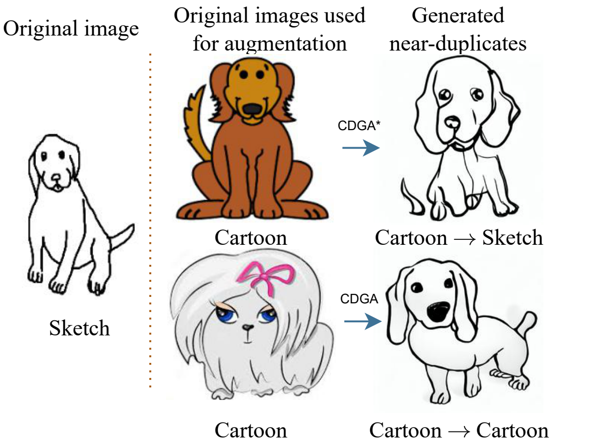


Figure 4: Examples of near-duplicates (right-most column) found for the dog image in Sketch domain (left-most column) that are generated using CDGA from the original images (middle column).

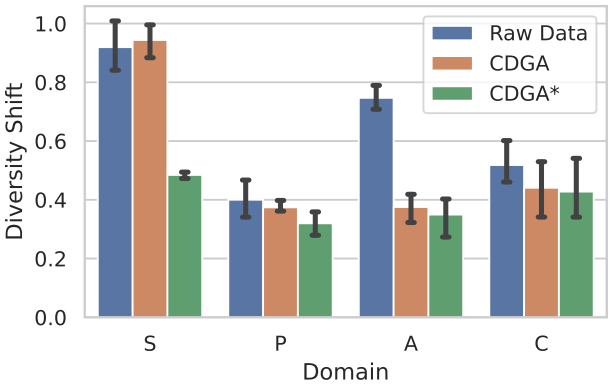


Figure 5: (left) Diversity shift as a measure of iid-ness of PACS (raw data) against CDGA, and CDGA* augmented datasets. Each column is the target domain and the rest of the domains are training domains. CDGA and CDGA* reduce the diversity shift.

within the manifold between training domains allows us to obtain examples that are near-duplicates of the target domain. Figure 4 showcases some of the near-duplicates identified for a sample image in the S domain using this technique. Additional examples can be found in Figure 8 in the Appendix B.

5. Conclusions

In this paper, we showed that a simple cross domain generative augmentation (i.e., CDGA) alongside ERM outperforms SOTA DG algorithms in the standard DomainBed benchmark. Empirically, using different distribution shift quantification techniques, we observe that the generated synthetic images from the vicinity distribution around each domain pair lead to a significant reduction in distribution shift between training domains after applying CDGA. Mitigating the distribution shift between domains reduces the estimation error of true data distribution in ERM which resulted in SOTA domain generalization of ERM along with CDGA. Our work provides a novel data-centric point of view for domain generalization, in the era of generative foundation models. For future work, we plan to dive into the possible theoretical implications of employing CDGA and the reasons behind the superior performance of this basic data generation technique compared to complex DG algorithms.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

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

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: 2178–2186, 2011.

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.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Fang, C., Xu, Y., and Rockmore, D. N. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1657–1664, 2013.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

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

Hemati, S., Zhang, G., Estiri, A., and Chen, X. Understanding hessian alignment for domain generalization. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Ilse, M., Tomczak, J. M., and Forré, P. Selecting data augmentation for simulating interventions. In *International Conference on Machine Learning*, pp. 4555–4562. PMLR, 2021.

Koyama, M. and Yamaguchi, S. When is invariance useful in an out-of-distribution generalization problem? *arXiv preprint arXiv:2008.01883*, 2020.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021.

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

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

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Parascandolo, G., Neitz, A., Orvieto, A., Gresele, L., and Schölkopf, B. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1406–1415, 2019.

Pinkney, J. Image mixer, 2023. URL `https://huggingface.co/lambdalabs/image-mixer`.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Rame, A., Dancette, C., and Cord, M. Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, pp. 18347–18377. PMLR, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Sener, O. and Koltun, V. Domain generalization without excess empirical risk. *Advances in Neural Information Processing Systems*, 35:13380–13391, 2022.

Shahtalebi, S., Gagnon-Audet, J.-C., Laleh, T., Faramarzi, M., Ahuja, K., and Rish, I. Sand-mask: An enhanced gradient masking strategy for the discovery of invariances in domain generalization. *arXiv preprint arXiv:2106.02266*, 2021.

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

Somavarapu, N., Ma, C.-Y., and Kira, Z. Frustratingly simple domain generalization via image stylization. *arXiv preprint arXiv:2006.11207*, 2020.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Sun, B. and Saenko, K. Deep CORAL: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450. Springer, 2016.

Tian, Y., Fan, L., Isola, P., Chang, H., and Krishnan, D. Stablerep: Synthetic images from text-to-image models make strong visual representation learners. *arXiv preprint arXiv:2306.00984*, 2023.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

Vapnik, V. N. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

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

Ye, N., Li, K., Bai, H., Yu, R., Hong, L., Zhou, F., Li, Z., and Zhu, J. Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7947–7958, 2022.

Zhang, G., Zhao, H., Yu, Y., and Poupart, P. Quantifying and improving transferability in domain generalization. *Advances in Neural Information Processing Systems*, 34: 10957–10970, 2021.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhang, L. and Agrawala, M. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.

Zhou, K., Yang, Y., Qiao, Y., and Xiang, T. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021.

A. Implementation Details

Models: We use the pretrained, version 1.4 of stable diffusion (Rombach et al., 2022) without finetuning as our base LDM. For the implementation of CDGA-IG, we use the image mixer that has been fine-tuned by Justin Pinkney at Lambda Labs (Pinkney, 2023) to accept CLIP image embeddings. For image generation, we do not tune any hyperparameters (e.g., strength, steps, etc) and all the parameters are set to their default values of the (Rombach et al., 2022) repository.

Prompts: For CDGA-PG, we use both the classes of the images and the domain description in the text prompts as guidance. The complete list of the prompts used for each domain in each dataset is in appendix E.

Implementation method: For implementing CDGA, we use offline augmentation where we first generate images between each pair of training domains (the workflow is illustrated in Figure 1), and then start the training process. The folder structure of our implementation for the PACS dataset when using P and A domains as train domains and S domain for test domain is illustrated in Figure 6. For all the methods, we set generation batch size $b = 1$ unless stated otherwise.

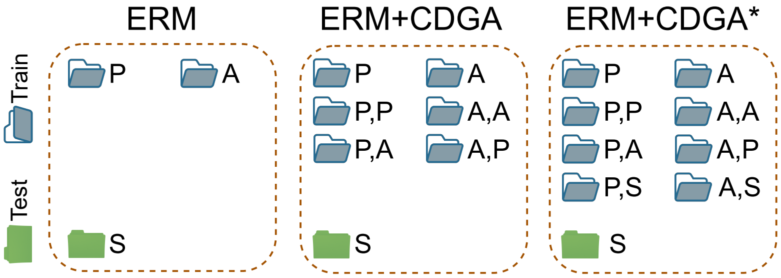


Figure 6: Illustration of the implementation structure of ERM, CDGA, and CDGA* on PACS dataset when using P and A domains as training and S as target domain.

Hardware: We use two clusters of four V100 NVIDIA GPUs for generation and benchmarks.

B. Test Domain Near-duplication Analysis Full Results

To quantify how much the generated images are similar to the original images for each domain, in section **??**, Figures 3 and 4 we presented the summarised results for near-duplicate image detection. More precisely, near-duplicate image detection was applied to images generated using CDGA to quantify how much the generated images are similar to the original images for each domain. Here, we present the extended version of these results in Figures 7 and 8 respectively. In Figure 7, for each original domain, we report the percentage of near-duplicates over the size of the original domain. Clearly, generating synthetic images that exist in the manifold between training domains enables us to have examples near-duplicate to the target domain. Figure 8 shows multiple examples where the synthetically generated images are near-duplicates to real data. These examples show how CDGA can reduce the domain shift between training domains and the target domain.

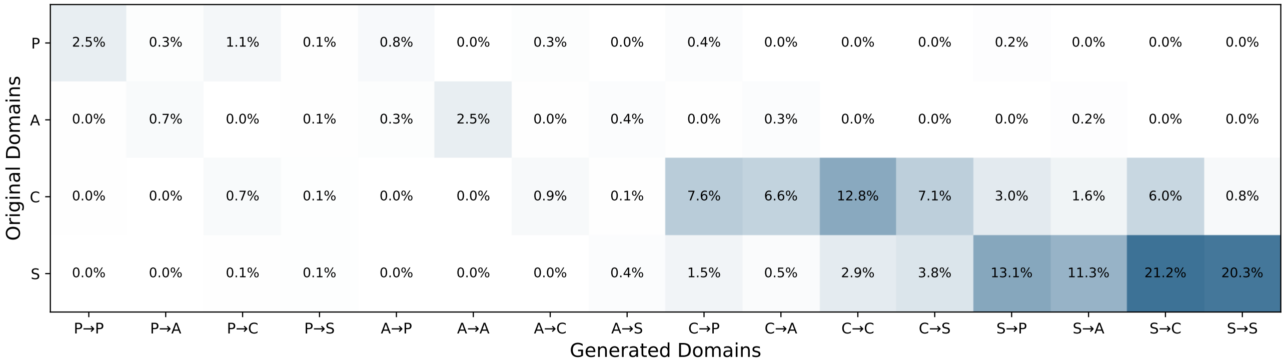


Figure 7: Heat map of the number of near-duplicates of each target domain that are in each original and generated dataset. This table shows that using test-domain description results in more near-duplicate images.

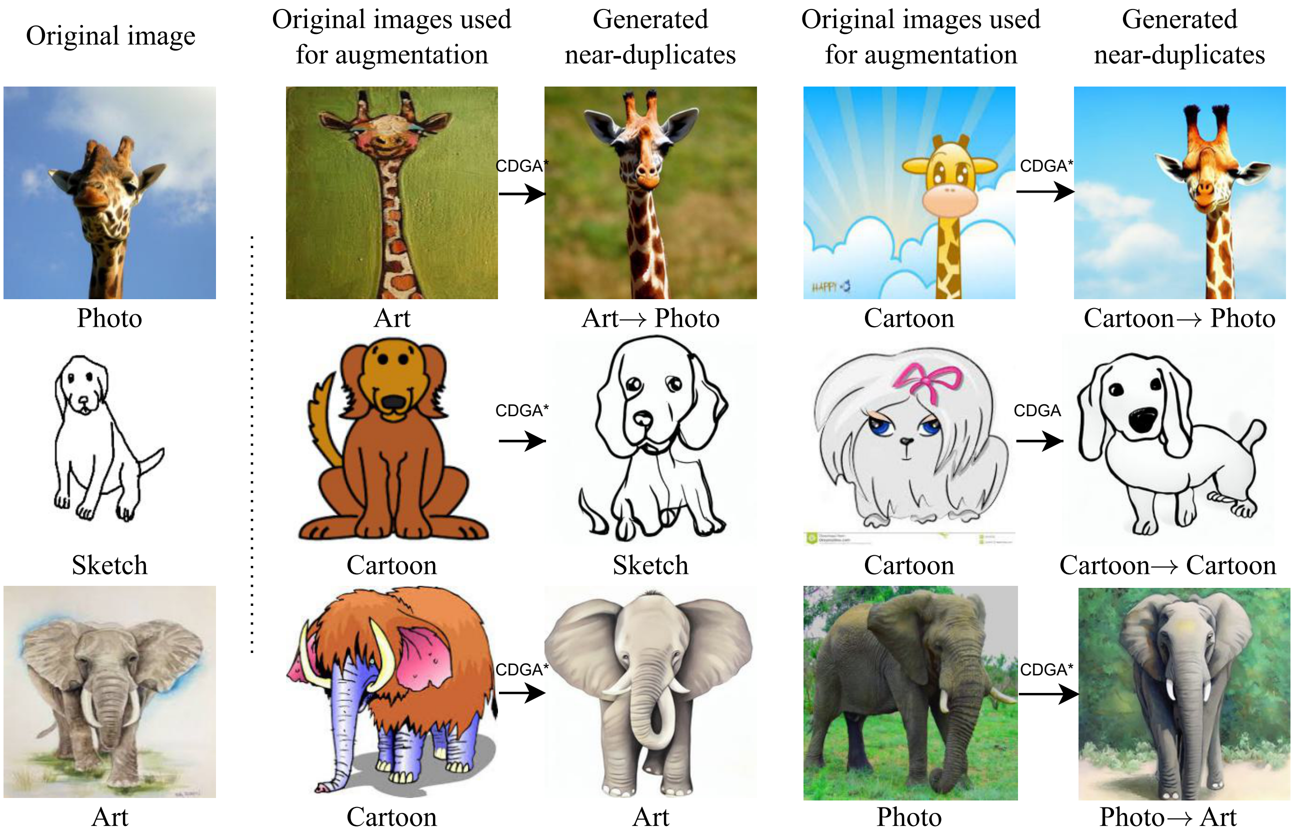


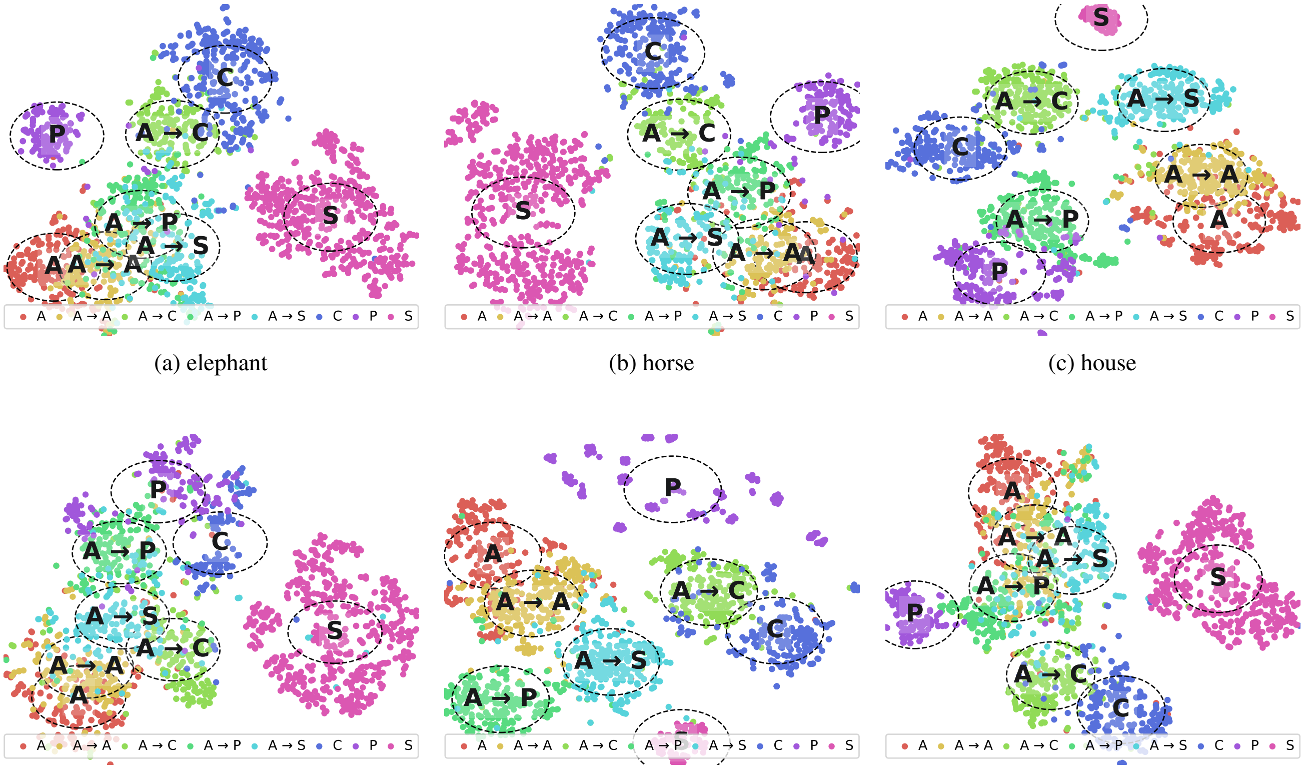
Figure 8: Illustration of near-duplicates of three images from the test domain (left-most column) that are generated using cross-domain generative augmentation (denoted by the arrow) from the original images and are in the training domain.

C. t-SNE plots

In Figure 2, we presented a 2D projection of the original PACS dataset from all domains along with CDGA-based data obtained from Domain A only for the “Dog" class. This figure showed how the cross-domain synthetic images interpolate different domains as we desired. Here in Figure 9, we present the results of this experiment for all other classes in the PACS dataset. As can be seen, for most classes the synthetic examples consistently reduce the domain shift which results in better OOD performance of ERM.

D. DomainBed benchmark full results

To save space in the main paper, for the DomainBed results in Tables 1- 4 we only reported the five top-performing methods for each model selection technique. Here in Tables 5, 6, 7, and 8 we present the results for all algorithms that have been tested on the DomainBed benchmark (Rame et al., 2022; Gulrajani & Lopez-Paz, 2020). Given that all the results presented for the DomainBed so far are averaged performances for the leave-one-domain-out experiments. The detailed per-domain results for PACS, OfficeHome, DomainNet, and VLCS are presented in Tables 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, and 20.



(a) elephant

(b) horse

(c) house

(d) guitar

(e) person

(f) giraffe

Figure 9: The t-SNE plot of features extracted from the original PACS dataset and generated images using CDGA by the LDM from A domain for all classes. This figure shows that CDGA can fill the gap between domains.

Table 5: DomainBed benchmark for **training-domain validation set** model selection method. We format **first**, second and worse than ERM results.

Algorithm	PACS	OfficeHome	DomainNet	Avg
ERM	85.5 ± 0.2	66.5 ± 0.3	40.9 ± 1.8	64.3
IRM	83.5 ± 0.8	64.3 ± 2.2	33.9 ± 2.8	60.6
GroupDRO	84.4 ± 0.8	66.0 ± 0.7	33.3 ± 0.2	61.2
Mixup	84.6 ± 0.6	68.1 ± 0.3	39.2 ± 0.1	64.0
MLDG	84.9 ± 1.0	66.8 ± 0.6	41.2 ± 0.1	64.3
CORAL	86.2 ± 0.3	68.7 ± 0.3	41.5 ± 0.1	65.5
MMD	84.6 ± 0.5	66.3 ± 0.1	23.4 ± 9.5	58.1
DANN	83.6 ± 0.4	65.9 ± 0.6	38.3 ± 0.1	62.6
CDANN	82.6 ± 0.9	65.8 ± 1.3	38.3 ± 0.3	62.2
MTL	84.6 ± 0.5	66.4 ± 0.5	40.6 ± 0.1	63.9
SagNet	86.3 ± 0.2	68.1 ± 0.1	40.3 ± 0.1	64.9
ARM	85.1 ± 0.4	64.8 ± 0.3	35.5 ± 0.2	61.8
V-REx	84.9 ± 0.6	66.4 ± 0.6	33.6 ± 2.9	61.6
RSC	85.2 ± 0.9	65.5 ± 0.9	38.9 ± 0.5	63.2
AND-mask	84.4 ± 0.9	65.6 ± 0.4	37.2 ± 0.6	62.4
SAND-mask	84.6 ± 0.9	65.8 ± 0.4	32.1 ± 0.6	60.8
Fish	85.5 ± 0.3	68.6 ± 0.4	42.7 ± 0.2	65.6
Fishr	85.5 ± 0.4	67.8 ± 0.1	41.7 ± 0.0	65.0
HGP	84.7 ± 0.0	68.2 ± 0.0	41.1 ± 0.0	64.7
Hutchinson	83.9 ± 0.0	68.2 ± 0.0	41.6 ± 0.0	64.6
ERM + CDGA-PG	88.5 ± 0.5	68.2 ± 0.6	43.1 ±0.0	66.6
ERM + CDGA-PG*	**89.5**± 0.3	**70.8** ± 0.6	**44.8** ±0.0	**68.4**

Table 6: DomainBed benchmark for **leave-one-domain-out cross-validation** model selection. We format **first**, second and worse than ERM results.

Algorithm	PACS	OfficeHome	DomainNet	Avg
ERM	83.0 ± 0.7	65.7 ± 0.5	40.6 ± 0.2	63.1
IRM	81.5 ± 0.8	64.3 ± 1.5	33.5 ± 0.3	59.8
GroupDRO	83.5 ± 0.2	65.2 ± 0.2	33.0 ± 0.3	60.6
Mixup	83.2 ± 0.4	67.0 ± 0.2	38.5 ± 0.3	62.9
MLDG	82.9 ± 1.7	66.1 ± 0.5	41.0 ± 0.2	63.3
CORAL	82.6 ± 0.5	68.5 ± 0.2	41.1 ± 0.1	64.1
MMD	83.2 ± 0.2	60.2 ± 5.2	23.4 ± 9.5	55.6
DANN	81.0 ± 1.1	64.9 ± 1.2	38.2 ± 0.2	61.4
CDANN	78.8 ± 2.2	64.3 ± 1.7	38.0 ± 0.1	60.4
MTL	83.7 ± 0.4	65.7 ± 0.5	40.6 ± 0.1	63.3
SagNet	82.3 ± 0.1	67.6 ± 0.3	40.2 ± 0.2	63.4
ARM	81.7 ± 0.2	64.4 ± 0.2	35.2 ± 0.1	60.4
V-REx	81.3 ± 0.9	64.9 ± 1.3	33.4 ± 3.1	59.9
RSC	82.6 ± 0.7	65.8 ± 0.7	38.9 ±0.5	62.4
HGP	82.2 ± 0.0	67.5 ± 0.0	41.1 ± 0.0	63.6
Hutchinson	84.8 ± 0.0	68.5 ± 0.0	41.4 ± 0.0	64.9
ERM + CDGA-PG	86.8 ±0.4	68.7 ± 0.4	43.1 ± 0.0	66.2
ERM + CDGA-PG*	**88.4** ±0.5	**70.2** ± 0.4	**44.8** ± 0.0	**67.8**

Table 7: DomainBed benchmark for **test-domain validation set (oracle)** model selection method. We format **first**, second and worse than ERM results.

Algorithm	PACS	OfficeHome	DomainNet	Avg
ERM	86.7 ± 0.3	66.4 ± 0.5	41.3 ± 0.1	64.8
IRM	84.5 ± 1.1	63.0 ± 2.7	28.0 ± 5.1	58.5
GroupDRO	87.1 ± 0.1	66.2 ± 0.6	33.4 ± 0.3	62.2
Mixup	86.8 ± 0.3	68.0 ± 0.2	39.6 ± 0.1	64.8
MLDG	86.8 ± 0.4	66.6 ± 0.3	41.6 ± 0.1	65.0
CORAL	87.1 ± 0.5	68.4 ± 0.2	41.8 ± 0.1	65.8
MMD	87.2 ± 0.1	66.2 ± 0.3	23.5 ± 9.4	59.0
DANN	85.2 ± 0.2	65.3 ± 0.8	38.3 ± 0.1	62.9
CDANN	85.8 ± 0.8	65.3 ± 0.5	38.5 ± 0.2	63.2
MTL	86.7 ± 0.2	66.5 ± 0.4	40.8 ± 0.1	64.7
SagNet	86.4 ± 0.4	67.5 ± 0.2	40.8 ± 0.2	64.9
ARM	85.8 ± 0.2	64.8 ± 0.4	36.0 ± 0.2	62.2
V-REx	87.2 ± 0.6	65.7 ± 0.3	30.1 ± 3.7	61.0
RSC	86.2 ± 0.5	66.5 ± 0.6	38.9 ± 0.6	63.9
AND-mask	86.4 ± 0.4	66.1 ± 0.2	37.9 ± 0.6	63.5
SAND-mask	85.9 ± 0.4	65.9 ± 0.5	32.2 ± 0.6	61.3
Fish	85.5 ± 0.3	68.6 ± 0.4	42.7 ± 0.2	65.6
Fishr	85.8 ± 0.6	66.0 ± 2.9	43.4 ± 0.3	65.1
Hutchinson	86.3 ± 0.0	68.4 ± 0.0	41.9 ± 0.0	65.5
HGP	86.5 ± 0.0	67.4 ± 0.0	41.2 ± 0.0	65.0
ERM + CDGA-PG	89.6 ± 0.3	68.8 ± 0.3	43.1 ±0.0	67.2
ERM + CDGA-PG*	**90.4** ± 0.3	**70.2** ± 0.2	**44.8** ±0.0	**68.5**

Table 8: DomainBed benchmark on **VLCS** dataset across different model selection methods. We format **first**, second and worse than ERM results.

Method	Training domain	Leave-one-domain-out	Oracle
ERM	77.5 ± 0.4	77.2 ± 0.4	77.6 ± 0.3
IRM	78.5 ± 0.5	76.3 ± 0.6	76.9 ± 0.6
GroupDRO	76.7 ± 0.6	77.9 ± 0.5	77.4 ± 0.5
Mixup	77.4 ± 0.6	77.7 ± 0.6	78.1 ± 0.3
MLDG	77.2 ± 0.4	77.2 ± 0.9	77.5 ± 0.1
CORAL	78.8 ± 0.6	78.7 ± 0.4	77.7 ± 0.2
MMD	77.5 ± 0.9	77.3 ± 0.5	77.9 ± 0.1
DANN	78.6 ± 0.4	76.9 ± 0.4	79.7 ± 0.5
CDANN	77.5 ± 0.1	77.5 ± 0.2	79.9 ± 0.2
MTL	77.2 ± 0.4	76.6 ± 0.5	77.7 ± 0.5
SagNet	77.8 ± 0.5	77.5 ± 0.3	77.6 ± 0.1
Fishr	77.8 ± 0.1	78.2 ± 0.0	78.2 ± 0.2
HGP	77.6 ± 0.0	76.7 ± 0.0	77.3 ± 0.0
Hutchinson	76.8 ± 0.0	**79.3** ± 0.0	77.9 ± 0.0
ERM + CDGA-IG	**78.9** ± 0.3	77.9 ± 0.5	**79.5** ± 0.1

Table 9: DomainBed benchmark, **PACS full results for training-domain validation set** model selection method.

Algorithm	A	C	P	S	Avg
ERM	84.7 ± 0.4	80.8 ± 0.6	97.2 ± 0.3	79.3 ± 1.0	85.5
IRM	84.8 ± 1.3	76.4 ± 1.1	96.7 ± 0.6	76.1 ± 1.0	83.5
GroupDRO	83.5 ± 0.9	79.1 ± 0.6	96.7 ± 0.3	78.3 ± 2.0	84.4
Mixup	86.1 ± 0.5	78.9 ± 0.8	97.6 ± 0.1	75.8 ± 1.8	84.6
MLDG	85.5 ± 1.4	80.1 ± 1.7	97.4 ± 0.3	76.6 ± 1.1	84.9
CORAL	88.3 ± 0.2	80.0 ± 0.5	97.5 ± 0.3	78.8 ± 1.3	86.2
MMD	86.1 ± 1.4	79.4 ± 0.9	96.6 ± 0.2	76.5 ± 0.5	84.6
DANN	86.4 ± 0.8	77.4 ± 0.8	97.3 ± 0.4	73.5 ± 2.3	83.6
CDANN	84.6 ± 1.8	75.5 ± 0.9	96.8 ± 0.3	73.5 ± 0.6	82.6
MTL	87.5 ± 0.8	77.1 ± 0.5	96.4 ± 0.8	77.3 ± 1.8	84.6
SagNet	87.4 ± 1.0	80.7 ± 0.6	97.1 ± 0.1	80.0 ± 0.4	86.3
ARM	86.8 ± 0.6	76.8 ± 0.5	97.4 ± 0.3	79.3 ± 1.2	85.1
V-REx	86.0 ± 1.6	79.1 ± 0.6	96.9 ± 0.5	77.7 ± 1.7	84.9
RSC	85.4 ± 0.8	79.7 ± 1.8	**97.6** ± 0.3	78.2 ± 1.2	85.2
AND-mask	85.3 ± 1.4	79.2 ± 2.0	96.9 ± 0.4	76.2 ± 1.4	84.4
SAND-mask	85.8 ± 1.7	79.2 ± 0.8	96.3 ± 0.2	76.9 ± 2.0	84.6
Fish	-	-	-	-	85.5
Fishr	88.4 ± 0.2	78.7 ± 0.7	97.0 ± 0.1	77.8 ± 2.0	85.5
CDGA-PG	89.1 ± 1.0	82.5 ± 0.5	97.4 ± 0.2	**84.8** ± 0.9	88.5
CDGA*-PG	**89.7** ± 1.1	**86.6** ± 0.3	97.4 ± 0.1	84.3 ± 1.6	**89.5**

Table 10: DomainBed benchmark, **PACS full results for leave-one-domain-out cross-validation** model selection method.

Algorithm	A	C	P	S	Avg
ERM	83.2 ± 1.3	76.8 ± 1.7	**97.2** ± 0.3	74.8 ± 1.3	83.0
IRM	81.7 ± 2.4	77.0 ± 1.3	96.3 ± 0.2	71.1 ± 2.2	81.5
GroupDRO	84.4 ± 0.7	77.3 ± 0.8	96.8 ± 0.8	75.6 ± 1.4	83.5
Mixup	85.2 ± 1.9	77.0 ± 1.7	96.8 ± 0.8	73.9 ± 1.6	83.2
MLDG	81.4 ± 3.6	77.9 ± 2.3	96.2 ± 0.3	76.1 ± 2.1	82.9
CORAL	80.5 ± 2.8	74.5 ± 0.4	96.8 ± 0.3	78.6 ± 1.4	82.6
MMD	84.9 ± 1.7	75.1 ± 2.0	96.1 ± 0.9	76.5 ± 1.5	83.2
DANN	84.3 ± 2.8	72.4 ± 2.8	96.5 ± 0.8	70.8 ± 1.3	81.0
CDANN	78.3 ± 2.8	73.8 ± 1.6	96.4 ± 0.5	66.8 ± 5.5	78.8
MTL	85.6 ± 1.5	78.9 ± 0.6	97.1 ± 0.3	73.1 ± 2.7	83.7
SagNet	81.1 ± 1.9	75.4 ± 1.3	95.7 ± 0.9	77.2 ± 0.6	82.3
ARM	85.9 ± 0.3	73.3 ± 1.9	95.6 ± 0.4	72.1 ± 2.4	81.7
VREx	81.6 ± 4.0	74.1 ± 0.3	96.9 ± 0.4	72.8 ± 2.1	81.3
RSC	83.7 ± 1.7	82.9± 1.1	95.6 ± 0.7	68.1 ± 1.5	82.6
CDGA-PG	87.3 ± 1.5	80.9 ± 1.6	96.6 ± 0.7	**82.5** ± 0.9	86.8
CDGA*-PG	**88.1** ± 1.1	**86.6** ± 1.0	**97.2** ± 0.4	81.9 ± 1.0	**88.4**

Table 11: DomainBed benchmark, **PACS full results for test-domain validation set (oracle)** model selection method.

Algorithm	A	C	P	S	Avg
ERM	86.5 ± 1.0	81.3 ± 0.6	96.2 ± 0.3	82.7 ± 1.1	86.7
IRM	84.2 ± 0.9	79.7 ± 1.5	95.9 ± 0.4	78.3 ± 2.1	84.5
GroupDRO	87.5 ± 0.5	82.9 ± 0.6	97.1 ± 0.3	81.1 ± 1.2	87.1
Mixup	87.5 ± 0.4	81.6 ± 0.7	97.4 ± 0.2	80.8 ± 0.9	86.8
MLDG	87.0 ± 1.2	82.5 ± 0.9	96.7 ± 0.3	81.2 ± 0.6	86.8
CORAL	86.6 ± 0.8	81.8 ± 0.9	97.1 ± 0.5	82.7 ± 0.6	87.1
MMD	88.1 ± 0.8	82.6 ± 0.7	97.1 ± 0.5	81.2 ± 1.2	87.2
DANN	87.0 ± 0.4	80.3 ± 0.6	96.8 ± 0.3	76.9 ± 1.1	85.2
CDANN	87.7 ± 0.6	80.7 ± 1.2	97.3 ± 0.4	77.6 ± 1.5	85.8
MTL	87.0 ± 0.2	82.7 ± 0.8	96.5 ± 0.7	80.5 ± 0.8	86.7
SagNet	87.4 ± 0.5	81.2 ± 1.2	96.3 ± 0.8	80.7 ± 1.1	86.4
ARM	85.0 ± 1.2	81.4 ± 0.2	95.9 ± 0.3	80.9 ± 0.5	85.8
V-REx	87.8 ± 1.2	81.8 ± 0.7	97.4 ± 0.2	82.1 ± 0.7	87.2
RSC	86.0 ± 0.7	81.8 ± 0.9	96.8 ± 0.7	80.4 ± 0.5	86.2
AND-mask	86.4 ± 1.1	80.8 ± 0.9	97.1 ± 0.2	81.3 ± 1.1	86.4
SAND-mask	86.1 ± 0.6	80.3 ± 1.0	97.1 ± 0.3	80.0 ± 1.3	85.9
Fish	-	-	-	-	85.8
Fishr	87.9 ± 0.6	80.8 ± 0.5	**97.9** ± 0.4	81.1 ± 0.8	86.9
CDGA-PG	89.6 ± 0.8	85.3 ± 0.7	97.3 ± 0.3	**86.2** ± 0.5	89.6
CDGA*-PG	**90.3** ± 0.8	**89.0** ± 0.2	96.8 ± 0.1	85.7 ± 1.0	**90.4**

Table 12: DomainBed benchmark, **OfficeHome full results for training-domain validation set** model selection method.

Algorithm	A	C	P	R	Avg
ERM	61.3 ± 0.7	52.4 ± 0.3	75.8 ± 0.1	76.6 ± 0.3	66.5
IRM	58.9 ± 2.3	52.2 ± 1.6	72.1 ± 2.9	74.0 ± 2.5	64.3
GroupDRO	60.4 ± 0.7	52.7 ± 1.0	75.0 ± 0.7	76.0 ± 0.7	66.0
Mixup	62.4 ± 0.8	54.8 ± 0.6	**76.9** ± 0.3	78.3 ± 0.2	68.1
MLDG	61.5 ± 0.9	53.2 ± 0.6	75.0 ± 1.2	77.5 ± 0.4	66.8
CORAL	**65.3** ± 0.4	54.4 ± 0.5	76.5 ± 0.1	78.4 ± 0.5	68.7
MMD	60.4 ± 0.2	53.3 ± 0.3	74.3 ± 0.1	77.4 ± 0.6	66.3
DANN	59.9 ± 1.3	53.0 ± 0.3	73.6 ± 0.7	76.9 ± 0.5	65.9
CDANN	61.5 ± 1.4	50.4 ± 2.4	74.4 ± 0.9	76.6 ± 0.8	65.8
MTL	61.5 ± 0.7	52.4 ± 0.6	74.9 ± 0.4	76.8 ± 0.4	66.4
SagNet	63.4 ± 0.2	54.8 ± 0.4	75.8 ± 0.4	78.3 ± 0.3	68.1
ARM	58.9 ± 0.8	51.0 ± 0.5	74.1 ± 0.1	75.2 ± 0.3	64.8
V-REx	60.7 ± 0.9	53.0 ± 0.9	75.3 ± 0.1	76.6 ± 0.5	66.4
RSC	60.7 ± 1.4	51.4 ± 0.3	74.8 ± 1.1	75.1 ± 1.3	65.5
ANDMask	59.5 ± 1.2	51.7 ± 0.2	73.9 ± 0.4	77.1 ± 0.2	65.6
SAND-mask	60.3 ± 0.5	53.3 ± 0.7	73.5 ± 0.7	76.2 ± 0.3	65.8
Fish	-	-	-	-	68.6
Fishr	62.4 ± 0.5	54.4 ± 0.4	76.2 ± 0.5	78.3 ± 0.1	67.8
CDGA-PG	60.1 ± 1.4	54.2 ± 0.5	78.2 ± 0.6	80.4 ± 0.1	68.2
CDGA*-PG	**63.1** ± 1.5	**60.2** ± 0.1	79.4 ± 0.7	**80.5** ± 0.2	**70.8**

Table 13: DomainBed benchmark, **OfficeHome full results for leave-one-domain-out cross-validation** model selection method.

Algorithm	**A**	**C**	**P**	**R**	**Avg**
ERM	61.1 ± 0.9	50.7 ± 0.6	74.6 ± 0.3	76.4 ± 0.6	65.7
IRM	58.2 ± 1.2	51.6 ± 1.2	73.3 ± 2.2	74.1 ± 1.7	64.3
GroupDRO	59.9 ± 0.4	51.0 ± 0.4	73.7 ± 0.3	76.0 ± 0.2	65.2
Mixup	61.4 ± 0.5	53.0 ± 0.3	75.8 ± 0.2	77.7 ± 0.3	67.0
MLDG	60.5 ± 1.4	51.9 ± 0.2	74.4 ± 0.6	77.6 ± 0.4	66.1
CORAL	64.5 ± 0.8	54.8 ± 0.2	76.6 ± 0.3	78.1 ± 0.2	68.5
MMD	60.8 ± 0.7	53.7 ± 0.5	50.2 ± 19.9	76.0 ± 0.7	60.2
DANN	60.2 ± 1.3	52.2 ± 0.9	71.3 ± 2.0	76.0 ± 0.6	64.9
CDANN	58.7 ± 2.9	49.0 ± 2.1	73.6 ± 1.0	76.0 ± 1.1	64.3
MTL	59.1 ± 0.3	52.1 ± 1.2	74.7 ± 0.4	77.0 ± 0.6	65.7
SagNet	**63.0** ± 0.8	54.0 ± 0.3	76.6 ± 0.3	76.8 ± 0.4	67.6
ARM	58.7 ± 0.8	49.8 ± 1.1	73.1 ± 0.5	75.9 ± 0.1	64.4
VREx	57.6 ± 3.4	51.3 ± 1.3	74.9 ± 0.2	75.8 ± 0.7	64.9
RSC	61.6 ± 1.0	51.1 ± 0.8	74.8 ± 1.1	75.7 ± 0.9	65.8
CDGA-PG	60.5 ± 1.2	56.5 ± 0.3	77.1 ± 0.4	**80.6** ± 0.2	68.7
CDGA*-PG	62.9 ± 0.4	**59.9** ± 0.5	**78.1** ± 0.9	79.9 ± 0.4	**70.2**

Table 14: DomainBed benchmark, **OfficeHome full results for test-domain validation set (oracle)** model selection method.

Algorithm	**A**	**C**	**P**	**R**	**Avg**
ERM	61.7 ± 0.7	53.4 ± 0.3	74.1 ± 0.4	76.2 ± 0.6	66.4
IRM	56.4 ± 3.2	51.2 ± 2.3	71.7 ± 2.7	72.7 ± 2.7	63.0
GroupDRO	60.5 ± 1.6	53.1 ± 0.3	75.5 ± 0.3	75.9 ± 0.7	66.2
Mixup	63.5 ± 0.2	54.6 ± 0.4	76.0 ± 0.3	78.0 ± 0.7	68.0
MLDG	60.5 ± 0.7	54.2 ± 0.5	75.0 ± 0.2	76.7 ± 0.5	66.6
CORAL	**64.8** ± 0.8	54.1 ± 0.9	76.5 ± 0.4	78.2 ± 0.4	68.4
MMD	60.4 ± 1.0	53.4 ± 0.5	74.9 ± 0.1	76.1 ± 0.7	66.2
DANN	60.6 ± 1.4	51.8 ± 0.7	73.4 ± 0.5	75.5 ± 0.9	65.3
CDANN	57.9 ± 0.2	52.1 ± 1.2	74.9 ± 0.7	76.2 ± 0.2	65.3
MTL	60.7 ± 0.8	53.5 ± 1.3	75.2 ± 0.6	76.6 ± 0.6	66.5
SagNet	62.7 ± 0.5	53.6 ± 0.5	76.0 ± 0.3	77.8 ± 0.1	67.5
ARM	58.8 ± 0.5	51.8 ± 0.7	74.0 ± 0.1	74.4 ± 0.2	64.8
V-REx	59.6 ± 1.0	53.3 ± 0.3	73.2 ± 0.5	76.6 ± 0.4	65.7
RSC	61.7 ± 0.8	53.0 ± 0.9	74.8 ± 0.8	76.3 ± 0.5	66.5
AND-mask	60.3 ± 0.5	52.3 ± 0.6	75.1 ± 0.2	76.6 ± 0.3	66.1
SAND-mask	59.9 ± 0.7	53.6 ± 0.8	74.3 ± 0.4	75.8 ± 0.5	65.9
Fish	-	-	-	-	66.0
Fishr	63.4 ± 0.8	54.2 ± 0.3	76.4 ± 0.3	78.5 ± 0.2	68.2
CDGA-PG	61.1 ± 1.1	55.9 ± 1.0	**78.2** ± 0.8	79.8 ± 0.2	68.5
CDGA*-PG	64.0 ± 0.2	**58.3** ± 0.4	77.7 ± 0.4	**80.8** ± 0.1	**70.2**

Table 15: DomainBed benchmark, **DomainNet full results for training-domain validation set** model selection method.

Algorithm	clip	info	paint	quick	real	sketch	Avg
ERM	58.1 ± 0.3	18.8 ± 0.3	46.7 ± 0.3	12.2 ± 0.4	59.6 ± 0.1	49.8 ± 0.4	40.9
IRM	48.5 ± 2.8	15.0 ± 1.5	38.3 ± 4.3	10.9 ± 0.5	48.2 ± 5.2	42.3 ± 3.1	33.9
GroupDRO	47.2 ± 0.5	17.5 ± 0.4	33.8 ± 0.5	9.3 ± 0.3	51.6 ± 0.4	40.1 ± 0.6	33.3
Mixup	55.7 ± 0.3	18.5 ± 0.5	44.3 ± 0.5	12.5 ± 0.4	55.8 ± 0.3	48.2 ± 0.5	39.2
MLDG	59.1 ± 0.2	19.1 ± 0.3	45.8 ± 0.7	**13.4** ± 0.3	59.6 ± 0.2	50.2 ± 0.4	41.2
CORAL	59.2 ± 0.1	19.7 ± 0.2	46.6 ± 0.3	**13.4** ± 0.4	59.8 ± 0.2	50.1 ± 0.6	41.5
MMD	32.1 ± 13.3	11.0 ± 4.6	26.8 ± 11.3	8.7 ± 2.1	32.7 ± 13.8	28.9 ± 11.9	23.4
DANN	53.1 ± 0.2	18.3 ± 0.1	44.2 ± 0.7	11.8 ± 0.1	55.5 ± 0.4	46.8 ± 0.6	38.3
CDANN	54.6 ± 0.4	17.3 ± 0.1	43.7 ± 0.9	12.1 ± 0.7	56.2 ± 0.4	45.9 ± 0.5	38.3
MTL	57.9 ± 0.5	18.5 ± 0.4	46.0 ± 0.1	12.5 ± 0.1	59.5 ± 0.3	49.2 ± 0.1	40.6
SagNet	57.7 ± 0.3	19.0 ± 0.2	45.3 ± 0.3	12.7 ± 0.5	58.1 ± 0.5	48.8 ± 0.2	40.3
ARM	49.7 ± 0.3	16.3 ± 0.5	40.9 ± 1.1	9.4 ± 0.1	53.4 ± 0.4	43.5 ± 0.4	35.5
V-REx	47.3 ± 3.5	16.0 ± 1.5	35.8 ± 4.6	10.9 ± 0.3	49.6 ± 4.9	42.0 ± 3.0	33.6
RSC	55.0 ± 1.2	18.3 ± 0.5	44.4 ± 0.6	12.2 ± 0.2	55.7 ± 0.7	47.8 ± 0.9	38.9
AND-mask	52.3 ± 0.8	16.6 ± 0.3	41.6 ± 1.1	11.3 ± 0.1	55.8 ± 0.4	45.4 ± 0.9	37.2
SAND-mask	43.8 ± 1.3	14.8 ± 0.3	38.2 ± 0.6	9.0 ± 0.3	47.0 ± 1.1	39.9 ± 0.6	32.1
Fish	-	-	-	-	-	-	42.7
Fishr	58.2 ± 0.5	20.2 ± 0.2	47.7 ± 0.3	12.7 ± 0.2	60.3 ± 0.2	50.8 ± 0.1	41.7
CDGA-PG	61.0 ± 0.2	20.2 ± 0.1	50.7 ± 0.1	11.1 ± 0.3	**65.3** ± 0.7	**54.0** ± 0.3	43.7
CDGA*-PG	**62.5** ± 0.0	**24.8** ± 0.0	**51.7** ± 0.0	11.7 ± 0.0	**65.2** ± 0.0	52.8 ± 0.0	**44.8**

Table 16: DomainBed benchmark, **DomainNet full results for leave-one-out** model selection method.

Algorithm	clip	info	paint	quick	real	sketch	Avg
ERM	58.1 ± 0.3	17.8 ± 0.3	47.0 ± 0.3	12.2 ± 0.4	59.2 ± 0.7	49.5 ± 0.6	40.6
IRM	47.5 ± 2.7	15.0 ± 1.5	37.3 ± 5.1	10.9 ± 0.5	48.0 ± 5.4	42.3 ± 3.1	33.5
GroupDRO	47.2 ± 0.5	17.0 ± 0.6	33.8 ± 0.5	9.2 ± 0.4	51.6 ± 0.4	39.2 ± 1.2	33.0
Mixup	54.4 ± 0.6	18.0 ± 0.4	44.5 ± 0.5	11.5 ± 0.2	55.8 ± 1.1	46.9 ± 0.2	38.5
MLDG	58.3 ± 0.7	19.3 ± 0.2	45.8 ± 0.7	13.2 ± 0.3	59.4 ± 0.2	49.8 ± 0.3	41.0
CORAL	59.2 ± 0.1	19.5 ± 0.3	46.2 ± 0.1	**13.4** ± 0.4	59.1 ± 0.5	49.5 ± 0.8	41.1
MMD	32.2 ± 13.3	11.0 ± 4.6	26.8 ± 11.3	8.7 ± 2.1	32.7 ± 13.8	28.9 ± 11.9	23.4
DANN	52.7 ± 0.1	18.0 ± 0.3	44.2 ± 0.7	11.8 ± 0.1	55.5 ± 0.4	46.8 ± 0.6	38.2
CDANN	53.1 ± 0.9	17.3 ± 0.1	43.7 ± 0.9	11.6 ± 0.6	56.2 ± 0.4	45.9 ± 0.5	38.0
MTL	57.3 ± 0.3	19.3 ± 0.2	45.7 ± 0.4	12.5 ± 0.1	59.3 ± 0.2	49.2 ± 0.1	40.6
SagNet	56.2 ± 0.3	18.9 ± 0.2	46.2 ± 0.5	12.6 ± 0.6	58.2 ± 0.6	49.1 ± 0.2	40.2
ARM	49.0 ± 0.7	15.8 ± 0.3	40.8 ± 1.1	9.4 ± 0.2	53.0 ± 0.4	43.4 ± 0.3	35.2
VREx	46.5 ± 4.1	15.6 ± 1.8	35.8 ± 4.6	10.9 ± 0.3	49.6 ± 4.9	42.0 ± 3.0	33.4
RSC	55.0 ± 1.2	18.3 ± 0.5	44.4 ± 0.6	12.2 ± 0.2	55.7 ± 0.7	47.8 ± 0.9	38.9
CDGA-PG	61.6 ± 0.1	20.6 ± 0.3	50.1 ± 0.4	11.2 ± 0.3	**64.5** ± 0.4	**53.8** ± 0.4	43.6
CDGA*-PG	**62.5** ± 0.0	**24.8** ± 0.0	**51.7** ± 0.0	11.7 ± 0.0	**65.2** ± 0.0	52.8 ± 0.0	**44.8**

Table 17: DomainBed benchmark, **DomainNet full results for test-domain validation set** (oracle) model selection method.

Algorithm	**clip**	**info**	**paint**	**quick**	**real**	**sketch**	**Avg**
ERM	58.6 ± 0.3	19.2 ± 0.2	47.0 ± 0.3	13.2 ± 0.2	59.9 ± 0.3	49.8 ± 0.4	41.3
IRM	40.4 ± 6.6	12.1 ± 2.7	31.4 ± 5.7	9.8 ± 1.2	37.7 ± 9.0	36.7 ± 5.3	28.0
GroupDRO	47.2 ± 0.5	17.5 ± 0.4	34.2 ± 0.3	9.2 ± 0.4	51.9 ± 0.5	40.1 ± 0.6	33.4
Mixup	55.6 ± 0.1	18.7 ± 0.4	45.1 ± 0.5	12.8 ± 0.3	57.6 ± 0.5	48.2 ± 0.4	39.6
MLDG	59.3 ± 0.1	19.6 ± 0.2	46.8 ± 0.2	13.4 ± 0.2	60.1 ± 0.4	50.4 ± 0.3	41.6
CORAL	59.2 ± 0.1	19.9 ± 0.2	47.4 ± 0.2	**14.0** ± 0.4	59.8 ± 0.2	50.4 ± 0.4	41.8
MMD	32.2 ± 13.3	11.2 ± 4.5	26.8 ± 11.3	8.8 ± 2.2	32.7 ± 13.8	29.0 ± 11.8	23.5
DANN	53.1 ± 0.2	18.3 ± 0.1	44.2 ± 0.7	11.9 ± 0.1	55.5 ± 0.4	46.8 ± 0.6	38.3
CDANN	54.6 ± 0.4	17.3 ± 0.1	44.2 ± 0.7	12.8 ± 0.2	56.2 ± 0.4	45.9 ± 0.5	38.5
MTL	58.0 ± 0.4	19.2 ± 0.2	46.2 ± 0.1	12.7 ± 0.2	59.9 ± 0.1	49.0 ± 0.0	40.8
SagNet	57.7 ± 0.3	19.1 ± 0.1	46.3 ± 0.5	13.5 ± 0.4	58.9 ± 0.4	49.5 ± 0.2	40.8
ARM	49.6 ± 0.4	16.5 ± 0.3	41.5 ± 0.8	10.8 ± 0.1	53.5 ± 0.3	43.9 ± 0.4	36.0
V-REx	43.3 ± 4.5	14.1 ± 1.8	32.5 ± 5.0	9.8 ± 1.1	43.5 ± 5.6	37.7 ± 4.5	30.1
RSC	55.0 ± 1.2	18.3 ± 0.5	44.4 ± 0.6	12.5 ± 0.1	55.7 ± 0.7	47.8 ± 0.9	38.9
AND-mask	52.3 ± 0.8	17.3 ± 0.5	43.7 ± 1.1	12.3 ± 0.4	55.8 ± 0.4	46.1 ± 0.8	37.9
SAND-mask	43.8 ± 1.3	15.2 ± 0.2	38.2 ± 0.6	9.0 ± 0.2	47.1 ± 1.1	39.9 ± 0.6	32.2
Fish	-	-	-	-	-	-	43.4
Fishr	58.3 ± 0.5	20.2 ± 0.2	47.9 ± 0.2	13.6 ± 0.3	60.5 ± 0.3	50.5 ± 0.3	41.8
CDGA-PG	61.6 ± 0.1	20.9 ± 0.2	51.8 ± 0.1	12.7 ± 0.2	**66.0** ± 0.5	**54.4** ± 0.2	44.4
CDGA*-PG	**62.5** ± 0.0	**24.8** ± 0.0	**51.7** ± 0.0	11.7 ± 0.0	**65.2** ± 0.0	52.8 ± 0.0	**44.8**

Table 18: DomainBed benchmark, **VLCS full results for training-domain validation set** model selection method.

Algorithm	**C**	**L**	**S**	**V**	**Avg**
ERM	97.7 ± 0.4	64.3 ± 0.9	73.4 ± 0.5	74.6 ± 1.3	77.5
IRM	98.6 ± 0.1	64.9 ± 0.9	73.4 ± 0.6	77.3 ± 0.9	78.5
GroupDRO	97.3 ± 0.3	63.4 ± 0.9	69.5 ± 0.8	76.7 ± 0.7	76.7
Mixup	98.3 ± 0.6	64.8 ± 1.0	72.1 ± 0.5	74.3 ± 0.8	77.4
MLDG	97.4 ± 0.2	**65.2** ± 0.7	71.0 ± 1.4	75.3 ± 1.0	77.2
CORAL	98.3 ± 0.1	66.1 ± 1.2	73.4 ± 0.3	**77.5** ± 1.2	78.8
MMD	97.7 ± 0.1	64.0 ± 1.1	72.8 ± 0.2	75.3 ± 3.3	77.5
DANN	**99.0** ± 0.3	65.1 ± 1.4	73.1 ± 0.3	77.2 ± 0.6	78.6
CDANN	97.1 ± 0.3	65.1 ± 1.2	70.7 ± 0.8	77.1 ± 1.5	77.5
MTL	97.8 ± 0.4	64.3 ± 0.3	71.5 ± 0.7	75.3 ± 1.7	77.2
SagNet	97.9 ± 0.4	64.5 ± 0.5	71.4 ± 1.3	**77.5** ± 0.5	77.8
ARM	98.7 ± 0.2	63.6 ± 0.7	71.3 ± 1.2	76.7 ± 0.6	77.6
V-REx	98.4 ± 0.3	64.4 ± 1.4	**74.1** ± 0.4	76.2 ± 1.3	78.3
RSC	97.9 ± 0.1	62.5 ± 0.7	72.3 ± 1.2	75.6 ± 0.8	77.1
AND-mask	97.8 ± 0.4	64.3 ± 1.2	73.5 ± 0.7	76.8 ± 2.6	78.1
SAND-mask	98.5 ± 0.3	63.6 ± 0.9	70.4 ± 0.8	77.1 ± 0.8	77.4
Fish	-	-	-	-	77.8
Fishr	98.9 ± 0.3	64.0 ± 0.5	71.5 ± 0.2	76.8 ± 0.7	77.8
CDGA-IG	96.3 ± 0.7	**75.7** ± 1.0	72.8 ± 1.3	73.7 ± 1.3	**79.6**

Table 19: DomainBed benchmark, **VLCS full results for eave-one-domain-out cross-validation** model selection.

Algorithm	**C**	**L**	**S**	**V**	**Avg**
ERM	98.0 ± 0.4	62.6 ± 0.9	70.8 ± 1.9	77.5 ± 1.9	77.2
IRM	**98.6** ± 0.3	66.0 ± 1.1	69.3 ± 0.9	71.5 ± 1.9	76.3
GroupDRO	98.1 ± 0.3	66.4 ± 0.9	71.0 ± 0.3	76.1 ± 1.4	77.9
Mixup	98.4 ± 0.3	63.4 ± 0.7	72.9 ± 0.8	76.1 ± 1.2	77.7
MLDG	98.5 ± 0.3	61.7 ± 1.2	**73.6** ± 1.8	75.0 ± 0.8	77.2
CORAL	96.9 ± 0.9	65.7 ± 1.2	73.3 ± 0.7	**78.7** ± 0.8	78.7
MMD	98.3 ± 0.1	65.6 ± 0.7	69.7 ± 1.0	75.7 ± 0.9	77.3
DANN	97.3 ± 1.3	63.7 ± 1.3	72.6 ± 1.4	74.2 ± 1.7	76.9
CDANN	97.6 ± 0.6	63.4 ± 0.8	70.5 ± 1.4	78.6 ± 0.5	77.5
MTL	97.6 ± 0.6	60.6 ± 1.3	71.0 ± 1.2	77.2 ± 0.7	76.6
SagNet	97.3 ± 0.4	61.6 ± 0.8	73.4 ± 1.9	77.6 ± 0.4	77.5
ARM	97.2 ± 0.5	62.7 ± 1.5	70.6 ± 0.6	75.8 ± 0.9	76.6
VREx	96.9 ± 0.3	64.8 ± 2.0	69.7 ± 1.8	75.5 ± 1.7	76.7
RSC	97.5 ± 0.6	63.1 ± 1.2	73.0 ± 1.3	76.2 ± 0.5	77.5
ERM+GA txt2im- label	96.5 ± 1.3	**75.4** ± 1.4	71.0 ± 2.4	78.1 ± 1.8	**80.3**

Table 20: **DomainBed benchmark**, VLCS full results for test-domain validation set (oracle) model selection method.

Algorithm	**C**	**L**	**S**	**V**	**Avg**
ERM	97.6 ± 0.3	67.9 ± 0.7	70.9 ± 0.2	74.0 ± 0.6	77.6
IRM	97.3 ± 0.2	66.7 ± 0.1	71.0 ± 2.3	72.8 ± 0.4	76.9
GroupDRO	97.7 ± 0.2	65.9 ± 0.2	72.8 ± 0.8	73.4 ± 1.3	77.4
Mixup	97.8 ± 0.4	67.2 ± 0.4	71.5 ± 0.2	75.7 ± 0.6	78.1
MLDG	97.1 ± 0.5	66.6 ± 0.5	71.5 ± 0.1	75.0 ± 0.9	77.5
CORAL	97.3 ± 0.2	67.5 ± 0.6	71.6 ± 0.6	74.5 ± 0.0	77.7
MMD	98.8 ± 0.0	66.4 ± 0.4	70.8 ± 0.5	75.6 ± 0.4	77.9
DANN	**99.0** ± 0.2	66.3 ± 1.2	73.4 ± 1.4	**80.1** ± 0.5	79.7
CDANN	98.2 ± 0.1	68.8 ± 0.5	**74.3** ± 0.6	78.1 ± 0.5	79.9
MTL	97.9 ± 0.7	66.1 ± 0.7	72.0 ± 0.4	74.9 ± 1.1	77.7
SagNet	97.4 ± 0.3	66.4 ± 0.4	71.6 ± 0.1	75.0 ± 0.8	77.6
ARM	97.6 ± 0.6	66.5 ± 0.3	72.7 ± 0.6	74.4 ± 0.7	77.8
V-REx	98.4 ± 0.2	66.4 ± 0.7	72.8 ± 0.1	75.0 ± 1.4	78.1
RSC	98.0 ± 0.4	67.2 ± 0.3	70.3 ± 1.3	75.6 ± 0.4	77.8
AND-mask	98.3 ± 0.3	64.5 ± 0.2	69.3 ± 1.3	73.4 ± 1.3	76.4
SAND-mask	97.6 ± 0.3	64.5 ± 0.6	69.7 ± 0.6	73.0 ± 1.2	76.2
Fish					77.8
Fishr	97.6 ± 0.7	67.3 ± 0.5	72.2 ± 0.9	75.7 ± 0.3	78.2
CDGA-IG	96.6 ± 0.7	**75.5** ± 1.9	73.6 ± 1.1	77.8 ± 1.0	**80.9**

E. Prompts

All prompts follow the same structure i.e., "a <class label>, <domain description>" where the domain descriptions for PACS, OfficeHome, and DomainNet are as follows:

E.1. PACS

- Photos: photorealistic, extremely detailed
- Sketches: sketch drawing, black and white, less details
- Cartoons: cartoon, cartoonish
- Art: art painting

E.2. OfficeHome

- Clipart: Clipart, schematic, simplified
- Product: Product, Merchandise
- Real: Real World, extremely detailed
- Art: art painting, art

E.3. Domainnet

- Clipart: cartoon, cartoonish, drawing
- Infograph: infographic, data visualization, poster
- Real: photorealistic, extremely detailed
- Painting: art painting
- Quickdraw: extremely simple drawing, black and white
- Sketch: sketch drawing, black and white, less details
- Clipart: cartoon, cartoonish, drawing

F. Code

To reproduce the DomainBed results, each class-specific dataset object inherits from either CDGA or CDGA* classes provided in this section. See the script provided in the section F.

```
class CDGA(MultipleDomainDataset):
    def __init__(self, root, test_envs, augment, hparams):
        super().__init__()

        transform = transforms.Compose(
            [
                transforms.Resize((224, 224)),
                transforms.ToTensor(),
                transforms.Normalize(
                    mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]
                ),
            ]
        )

        augment_transform = transforms.Compose(
            [
                # transforms.Resize((224,224)),
                transforms.RandomResizedCrop(224, scale=(0.7, 1.0)),
                transforms.RandomHorizontalFlip(),
                transforms.ColorJitter(0.3, 0.3, 0.3, 0.3),
                transforms.RandomGrayscale(),
                transforms.ToTensor(),
                transforms.Normalize(
                    mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]
                ),
            ]
        )

        environments = [f.name for f in os.scandir(root) if f.is_dir()]
        environments = sorted(environments)

        self.datasets = []
        print(f"Test domains: {test_envs}")
        for i, environment in enumerate(environments):
            # Transformation
            if augment and (i not in test_envs):
                env_transform = augment_transform
            else:
                env_transform = transform

            path = os.path.join(root, environment)
            # Create list of generated subfolders for each distribution
            sub_environments = [f.name for f in os.scandir(path) if f.is_dir()]
            if i not in test_envs:
                # if we are in the training distribution combine folders that are not in the test distributions
                env_dataset = []
                for sub_env in sub_environments:
                    if all(environments[i] not in sub_env for i in test_envs):
                        print(f"Adding {sub_env} to {environment} for training")
                        env_dataset.append(
                            ImageFolder(
                                os.path.join(path, sub_env), transform=env_transform
                            )
                        )
                self.datasets.append(torch.utils.data.ConcatDataset(env_dataset))
            else:
                # if we are in the testing distribution just use the original data
                print(f"using {environment} for testing")
                self.datasets.append(
                    ImageFolder(
                        os.path.join(path, environment), transform=env_transform
                    )
                )
        self.input_shape = (
            3,
            224,
            224,
        )


class CDGA_star(MultipleDomainDataset):
    def __init__(self, root, test_envs, augment, hparams):
        super().__init__()

        transform = transforms.Compose(
            [
                transforms.Resize((224, 224)),
                transforms.ToTensor(),
                transforms.Normalize(
                    mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]
                ),
            ]
        )

        augment_transform = transforms.Compose(
            [
                # transforms.Resize((224,224)),
                transforms.RandomResizedCrop(224, scale=(0.7, 1.0)),
                transforms.RandomHorizontalFlip(),
                transforms.ColorJitter(0.3, 0.3, 0.3, 0.3),
                transforms.RandomGrayscale(),
                transforms.ToTensor(),
                transforms.Normalize(
                    mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]
                ),
            ]
        )

        environments = [f.name for f in os.scandir(root) if f.is_dir()]
        environments = sorted(environments)

        self.datasets = []
        print(f"Test domains: {test_envs}")
        for i, environment in enumerate(environments):
            if augment and (i not in test_envs):
                env_transform = augment_transform
            else:
                env_transform = transform
            path = os.path.join(root, environment)
            # create list of generated subfolders for each distribution
            sub_environments = [f.name for f in os.scandir(path) if f.is_dir()]
            if i not in test_envs:
                # if we are in the training distribution combine all the test folder except the original test data
                env_dataset = []
                for sub_env in sub_environments:
                    print(f"Adding {sub_env} to {environment} for training")
                    env_dataset.append(
                        ImageFolder(
                            os.path.join(path, sub_env), transform=env_transform
                        )
                    )
                self.datasets.append(torch.utils.data.ConcatDataset(env_dataset))
            else:
                # if we are in the testing distribution just use the original data
                print(f"using {environment} for testing")
                self.datasets.append(
                    ImageFolder(
                        os.path.join(path, environment), transform=env_transform
                    )
                )
        self.input_shape = (
            3,
            224,
            224,
        )


class G_PACS(CDGA):
    CHECKPOINT_FREQ = 300
    ENVIRONMENTS = ["A", "C", "P", "S"]
    num_classes = 7

    def __init__(self, root, test_envs, hparams):
        self.dir = os.path.join(root, "G_PACS/")
        super().__init__(self.dir, test_envs, hparams["data_augmentation"], hparams)
```

G. Mitigating Class Imbalance

CDGA can also be utilized to mitigate the class imbalance problem in datasets where the number of instances in each class of each domain is not equal. In such scenarios, one can use a different b for each class of the data such that after generating samples, the number of instances in each class of generated domains becomes equal. We test the effectiveness of CDGA method in balancing the OfficeHome dataset (which is highly imbalanced) through the DomainBed benchmark. More specifically, for every class c and domain S_j, we find the number of samples $n(S_j, c)$ and then we find $m = \max_{c,j}\ n(S_j, c)$ which is 100 for OfficeHome. Then for every domain S_j and class c we set $b = \frac{m}{n(S_j,c)}$ which leads to larger batch size for domains and classes with fewer data points and subsequently balances the dataset. The results of this experiment are presented in Table 21. Clearly, by choosing b in a way that the dataset is more balanced, the OOD generalization has been further improved.

Table 21: OOD accuracy of models with and without balanced generation in OfficeHome dataset .

Method	Training domain	Leave-one -domain-out	Oracle
ERM	66.5 ± 0.3	65.7 ± 0.5	66.4 ± 0.5
ERM + CDGA ($b = 1$)	68.2 ± 0.6	68.7 ± 0.4	68.6 ± 0.3
ERM + CDGA ($b = \frac{m}{n(\mathcal{E}_j,c)}$)	**69.9** ± 0.2	**69.7** ± 0.4	**70.0** ± 0.7