SUPPLEMENTARY MATERIAL: DOMAINFUSION: GENERALIZING TO UNSEEN DOMAINS WITH LATENT DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

A APPENDIX

A.1 COMPARISON WITH SCORE DISTILLATION SAMPLING AND DIFFUSION CLASSIFIER

Comparison with Score Distillation Sampling. (1) GSD extends SDS paradigm to discriminative models and perception tasks. SDS is inherently restricted to cases where the targeted model is a generative model, thus limiting its applicability to tasks beyond generation. In contrast, Our GSD employs diffusion-like images as intermediaries to establish a connection between the parameter spaces of discriminative and latent diffusion models, facilitating the transfer of semantic knowledge for discriminative tasks. (2) GSD provides clearer evidence of its effectiveness. Equation **??** yields a compelling conclusion that GSD can be employed to optimize the KL divergence between the prediction distributions of the latent diffusion model and the DG network. This implies that GSD can provide supervision signals for the DG network similar to ground truth in supervised learning.

Comparison with Diffusion classifier. Empirically and experimentally, we find that the diffusion classifier(**??**) that directly uses noise for classification does not yield satisfactory results. The key reason behind this is that the diffusion classifier requires matching the correct image with a fake category and predicting the probability of this fake match. For example, using a picture of a dog and the text promt 'cat', the diffusion classifier is expected to provide the probability of the dog picture belonging to the cat category. However, the dog picture does not contain any information about cats. Consequently, utilizing incorrectly matched image-text pairs leads to noisy and inaccurate predictions. In contrast, our GSD approach **merely utilizes correctly matched image-text pairs**, effectively eliminating noisy predictions. Figure **??** illustrates the results of visualizing the diffusion classifier's score vectors by cross-attention map in the UNet obtained by DAAM(**?**). Images from Office Home are sequentially matched with a real label prompt and fake label prompts to compute the cross-attention map. It can be observed that diffusion fails to comprehend mismatched image-text pairs, resulting in unreliable predictions in such cases.

A.2 EXPERIMENTAL SETTINGS

Settings and Datasets. Following DomainBed, we conducted a series of experiments on five prominent real-world benchmark datasets: PACS(4 domains, 9,991 samples, and 7 classes), VLCS(4 domains, 10,729 samples, and 5 classes), OfficeHome(4 domains, 15,588 samples, and 65 classes), TerraIncognita(4 domains, 24,778 samples, and 10 classes), and DomainNet(6 domains, and 586,575 samples, and 345 classes). To ensure a fair and consistent comparison, we follow DomainBed's(**?**) established training and evaluation protocol. In this protocol, we designate one domain as the target, while the remaining domains serve as source domains. Model selection is conducted using the training-domain validation approach, where 20% of the source domain data is used for validation. The performance of domain generalization is evaluated individually on each domain and then averaged across all domains.

Implementation Details. For the latent diffusion model, we employ the stable diffusion v1-4 model card. Specifically, we utilize the image-to-image pipeline for image generation and loss extraction, where the input image size is set to 320x320, which greatly boosts algorithm training speed and reduces computational overhead, and other hyperparameters are set to their default values as specified by stable diffusion. For domain generalization, we utilize ResNet-50 pretrained on ImageNet and RegNet-Y-16GF pretrained using SWAG as our backbone models. The batch size is set to 16,

Table 1: Effects of Different Components in DomainFusion

$\mathcal{L}_{\text{raw}}$	$\mathcal{L}_{\text{gen}}$	$\mathcal{L}_{\text{GSD}}$	Art	Clipart	Product	Real	Avg.
✓	✗	✗	69.3	61.3	81.6	82.5	73.7
✓	✓	✗	73.6	71.2	80.7	88.7	78.6
✓	✓	✓	**81.2**	**73.9**	**88.5**	**90.1**	**83.4**

Table 2: Effects of the Sampling Strategy.

w/o	Art	Clipart	Product	Real	Avg.
✗	79.4	71.8	87.5	88.2	81.7
✓	**81.2**	**73.9**	**88.5**	**90.1**	**83.4**

Table 3: Effects of the Candidate Number.

candidate number	Art	Clipart	Product	Real	Avg.
$N = 1$	79.4	71.8	87.5	88.2	81.7
$N = 2$	**81.2**	**73.9**	**88.5**	**90.1**	**83.4**
$N = 5$	80.4	72.7	87.8	89.3	82.6

Table 4: Time cost hours of different components.

Algorithm	Clipart	Info	Painting	Quickdraw	Real	Sketch	Avg.
Diffusion Classifier	5.8	6.3	8.9	20.3	20.7	8.3	11.7
DomainFusion without GSD	21.1	17.4	18.5	17.8	17.4	17.2	18.2
DomainFusion with GSD	28.2	25.2	26.0	25.4	25.2	25.2	25.9

except for DomainNet where it is reduced to 8 due to computational limitations. We employ the Adam optimizer and cosine learning rate schedule during training.

A.3 Ablation study

We conduct experiments on Office Home for ablation study. All models are based on RegNet-Y-16GF and trained for 120 epochs.

Effects of Different Components. As shown in Table 1, $\mathcal{L}_{\text{gen}}$ improves the average accuracy by 4.9% by generating a more diverse set of samples to augment the source domain, resulting in a significant improvement in DG performance . However, using $\mathcal{L}_{\text{gen}}$ alone still exhibits a considerable performance gap compared to state-of-the-art methods. To address this discrepancy, $\mathcal{L}_{\text{GSD}}$ bridges this gap by further enhancing the accuracy by 4.8% compared to use $\mathcal{L}_{\text{gen}}$ solely.

Effects of the Sampling Strategy. Table 2 demonstrates the effect of the sampling strategy. The inclusion of the sampling strategy led to a significant enhancement of 1.7% in accuracy compared to the exclusion version, thereby indicating the effectiveness of the sampling strategy. The implementation of the sampling strategy allows for the optimization of both semantic and non-semantic factors, resulting in the generation of samples that are better aligned with the requirements of DG.

Effects of the Candidate Number. Table 3 presents the impact of the number of candidates, denoted as N, on the results. We considered three scenarios: $N = 1$, $N = 2$, and $N = 5$, with $N = 2$ being the default setting for DomainFusion. In the implementation process, N is primarily adjusted by the number of images generated for each prompt in the stable diffusion pipeline. It is noteworthy that a larger value of N may yield a decline performance because too many candidates may lead to visual clutter in the synthesized images. Therefore, setting N as 2 is deemed as a favorable choice.

A.4 cost analysis

We analyze the GPU time consumption of different components in DomainFusion on DomainNet, along with the runtime of the Diffusion Classifier for comparison. It is worth noting that all the reported times refer to the number of hours the algorithms consumed on 8* V100 GPUs. DomainFusion was run for 120 epochs and completed both training and inference, while the Diffusion Classifier only completed the inference phase. Despite the longer runtime of DomainFusion compared to the Diffusion Classifier, it remains affordable while achieving a significant improvement in accuracy. Note that when used for inference, our DomainFusion requires no extra time compared with ERM.

A.5 More visualization results

We present more visualization results of autoregressively generated samples and corresponding GSD noise images in Figure 1.

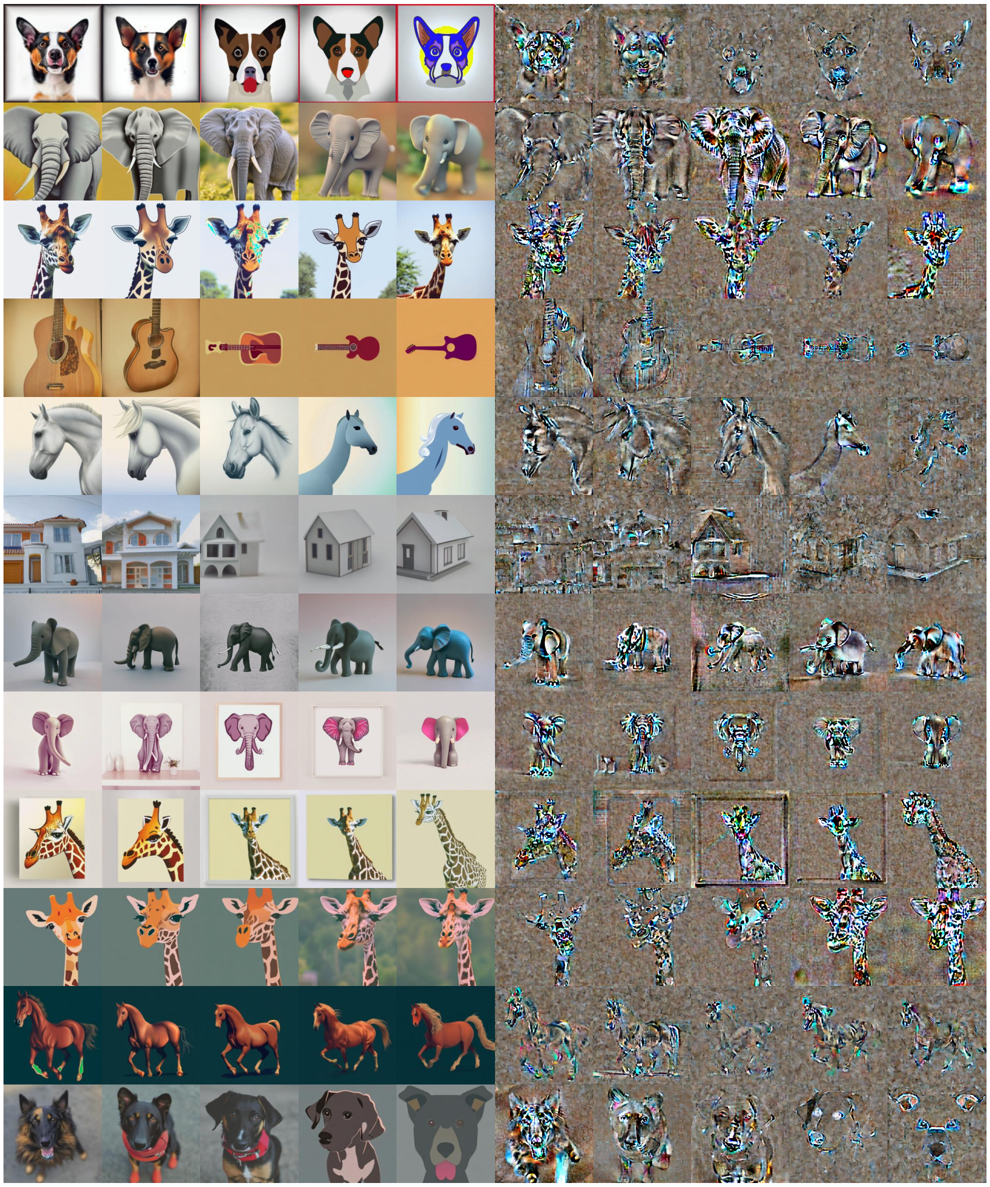

Figure 1: More visualization results of generated samples and GSD noise, with the left section being autoregressively generated samples and the right section being corresponding GSD noise.