Customizing Domain Adapters for Domain Generalization

Supplementary Material

More experimental details We adopt the experimental setup introduced in [14, 63] as our framework for evaluating our domain generalization approach. This framework entails maintaining consistency in various aspects, including model selection criteria, dataset partitioning, and the utilization of the same network backbone to ensure comparability. In alignment with the methodology employed in DomainBed, we opt for the ViT-S/16 model, as recommended in [27]. Notably, this choice is motivated by its proximity to ResNet-50 in terms of parameter count and runtime memory usage. For our comparative analysis, we benchmark our approach against the most recent and top-performing domain generalization algorithms. We compare with both ViT-Small/16 (with 21.7 million parameters) and ResNet50 (with 25.6 million parameters), both pre-trained on ImageNet-1k, as our backbone models.

Analysis of the domain adapters layer configuration Two prevalent block placement methods, often referenced in existing research [46, 50], are "every-two" and "last-two". "Every-two" involves situating the block in the even-numbered blocks, while "last-two" places the block in the final two even blocks, as illustrated in Figure 4 (a). Our empirical findings, presented in Table 10, indicate that "every-two" outperforms "last-two" in the context of our method.

Method	Layer Config	Acc (PACS)
Domain Adapter	Every 2	89.4 ± 0.3
Domain Adapter	Last 2	88.4 ± 0.2

Table 10. Analysis of layer configuration on PACS benchmark. For simplicity, we fix the domain adapter positioned after the MSA module and outside the residual connection of MSA here. "every-two" achieves better performance than "last-two" in our method.

Analysis of domain adapters position in ViT block We consider five different configurations of domain adapters within a vision transformer block, see Figure 4 (b). We empirically find that the first case achieves the best performance in Table 11.

Method	Position Config	Acc (PACS)
Domain Adapter	First case	89.4 ± 0.3
Domain Adapter	Second case	88.9 ± 0.3
Domain Adapter	Third case	88.6 ± 0.1
Domain Adapter	Fourth case	84.5 ± 0.4
Domain Adapter	Fifth case	83.6 ± 0.5

Table 11. Analysis of domain adapter position in ViT block on PACS benchmark. We fix layer configuration as "every 2" here. The first case achieves the best performance.

Result of domain adapter customization with GPT guidance. Similar to the tried-and-tested design choices in Table 8, GPT-guided adapter selection follows a similar trend in Table 12. Domains with limited color information, such as the sketch domain in the PACS dataset, are typically assigned ViT adapters, while more colorful domains, like photo and art, are matched with Conv adapters.

Dataset	Domain 1	Domain 2	Domain 3	Domain 4	Domain 5	Domain 6
PACS	photo	art	cartoon	sketch	-	-
adapter	Conv	Conv	Conv	ViT	-	-
VLCS	VOC	LABEL	CAL	SUN	-	-
adapter	Conv	Conv	ViT	ViT	-	-
OfficeHome	clip	art	real	product	-	-
adapter	ViT	ViT	ViT	ViT	-	-
Terra	L38	L43	L46	L100	-	-
adapter	Conv	Conv	Conv	Conv	-	-
DomainNet	clip	Info	paint	quick	real	sketch
adapter	ViT	Conv	Conv	ViT	Conv	ViT

Table 12. GPT guided domain adapter customization for five benchmarks.

Fully Finetune vs Only Adapter In Table 13, we found that fully fine-tuning all network parameters leads to better performance than fine-tuning only the custom domain adapters and classifier, with a 1.2% improvement on the PACS dataset and a 3.0% improvement on the OfficeHome dataset. This improvement likely arises from the relatively small size of our pre-trained network, where fine-tuning only the adapter may not be adequate for downstream tasks. As a result, we adopt comprehensive fine-tuning strategies in all our experiments.

Method	Acc(OfficeHome)	Acc(PACS)
Adapter & Classifier	73.1 ± 0.0	88.2 ± 0.2
Fully Finetune	76.1 ± 0.1	89.4 ± 0.3

Table 13. Analysis of Fully Finetune vs Only Adapter & Classifier.

The robustness of our method to noisy domain IDs In this section, we investigated the robustness of our CDA method to noisy domain IDs, as detailed in Table 14. Specifically, during training, we randomized 20% of the domain labels, meaning a domain-1 sample could be randomly reassigned to domain-2, and a domain-2 sample to domain-3, for example. The results show a slight degradation in performance when 20% of the domain labels are randomized, demonstrating CDA's resilience to such noise.

Compare with more methods In this section, we compare our method with the previous SOTA approach using the same ViT-S/16 backbone. Specifically, we implemented PCL [59] within the same ViT-S/16 architecture. The results, achieving

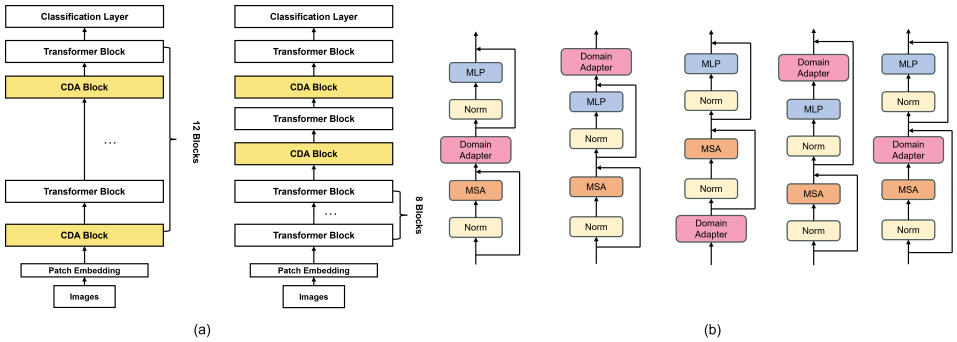

Figure 4. (a) Illustration of two different CDA layer configurations. The left: “every-two" placement. The right: “last-two" placement. (b) Illustration of five different configurations of a domain adapter within a vision transformer block. From left to right, the configurations are sequentially ordered from the first to the fifth case.

Method	Acc(OfficeHome)	Acc(PACS)
Original	76.1 $\pm$ 0.1	89.4 $\pm$ 0.3
20% domain label randomized	75.9 $\pm$ 0.1	88.8 $\pm$ 0.4

Table 14. Analysis of robustness to noisy domain IDs.

74.4±0.3 on OfficeHome and 48.5±0.2 on DomainNet, fall short of those obtained by our CDA method in Table 15.

Method	Backbone	Acc(OfficeHome)	Acc(DomainNet)
PCL[59]	ViT-S/16	74.4 $\pm$ 0.3	48.5 $\pm$ 0.2
SWAD [6]	ViT-S/16	73.8 $\pm$ 0.2	48.3 $\pm$ 0.3
GMoE[27]	ViT-S/16	74.2 $\pm$ 0.4	48.7 $\pm$ 0.2
CDA(Ours)	ViT-S/16	76.1 $\pm$ 0.1	50.3$\pm$ 0.4

Table 15. Compare with PCL [59] and SWAD [6] with the same ViT-S/16 backbone.

Experiments Compute Resources All experiments were conducted on an NVIDIA GeForce RTX 3090 graphics card with 24 GB of memory. Each individual experiment took approximately 30 minutes to run, with the total computation time for the entire project amounting to approximately 30 hours on the GPU.

5.1. Multi-head Attention Visualization

To gain deeper insights into how our method contributes to enhancing the domain generalization capabilities of the ViT model, we have visualized the attention maps for the last block of the ViT both with and without the integration of our method. These visualizations are presented in Figure 5 and Figure 6.

Our observations reveal that our method, the Customized Domain Adapters (CDA), tends to produce more comprehensive attention maps. This phenomenon could be attributed to using both ViT and CNN adapters within our model architecture. This combination potentially allows the model to leverage a broader range of information when making predictions, including more details such as the texture, color, and shape of objects.

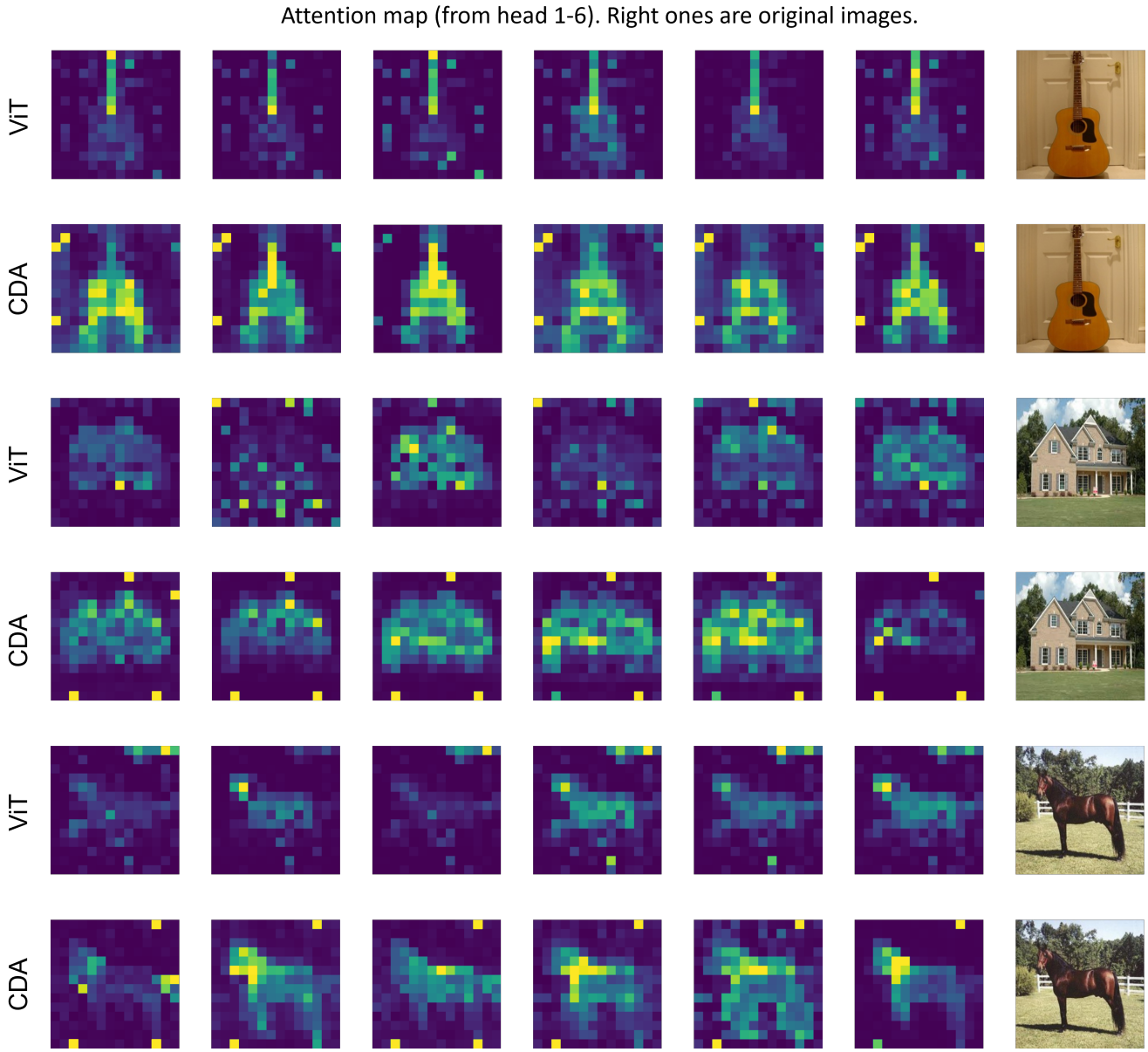

Figure 5. Multi-head attention visualization on the last block of ViT-S/16 and CDA-S/16. Images are from Photo domain in PACS.

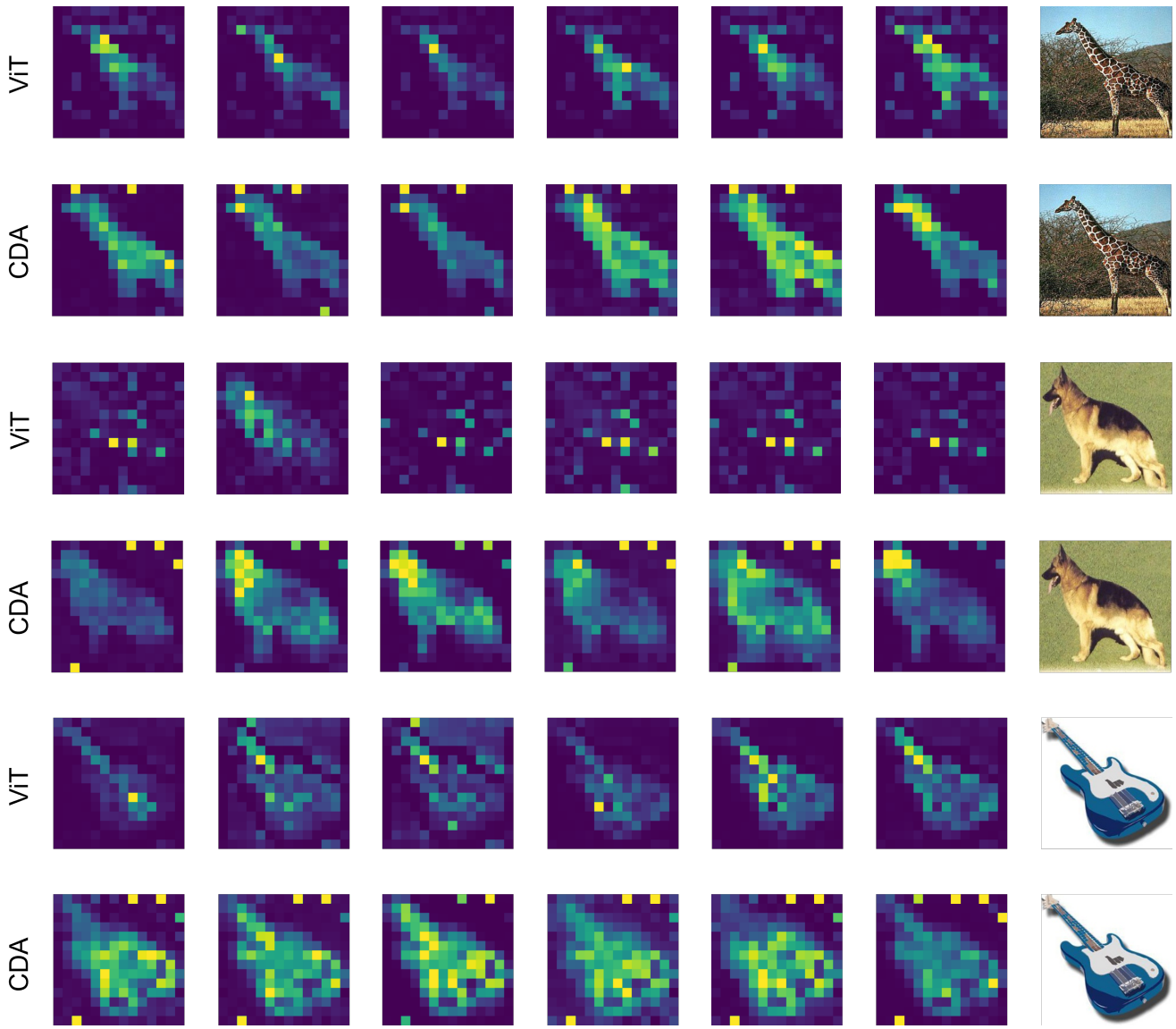

Figure 6. Multi-head attention visualization on the last block of ViT-S/16 and CDA-S/16. Images are from Photo domain in PACS.