Supplementary Material for DiPrompT: Disentangled Prompt Tuning for Multiple Latent Domain Generalization in Federated Learning

A. Pseudo Code for DiPrompT

We present the pseudo-code for DiPrompT during training time, where Q-Prompt as well as G-Prompts and D-Prompts are alternatively updated in each local client, while G-Prompts and D-Prompts are aggregated in server-side with different manners. The pseudo-code can be found in Algorithm 1

B. Dataset

VLCS [1] is another public image classification dataset, which aggregates instances from VOC2007, LabelMe, Caltech10, and SUN09 as 4 separated domains. It contains 10, 729 instances with 5 classes. Similar to PACS and Office-home, we adopt the "leave-one-domain-out" strategy and implement training in divided training examples from three domains and testing in the test set of the rest domain.

PACS [2] is a dataset containing 9,991 RGB images of 227x227 resolutions. It consists of four domains, namely Sketch (3,929 images), Cartoon (2,344 images), Art Painting (2,048 images), and Photo (1,670 images). Each domain contains seven categories: dog, elephant, giraffe, guitar, horse, house, and person. We follow the official guidance to split training and test data. In our DiPrompT training, we chose three domains for training and allocate their training data to K clients, where the training data from a single domain can spread to multiple clients but a client only contains data from a single domain. Moreover, the global model tests in the test set of the rest subset.

Officehome [3] is a larger image classification dataset, which contains around 15,500 images. It consists of images from 4 different domains: Artistic, Clip Art, Product images, and Real-World. For each domain, the sub-dataset contains images of 65 object categories found typically in office and home scenarios, such as hammer, chair, bike, table, knife and etc. Moreover, we split the training-test subset while assigning them to sever and different clients in Officehome with a similar manner to PACS.

C. Addtional Ablation Study

C.1. Domain Distance.

To validate the significant differences in domain shift among various domains, we describe the distances between different domains by the average inter-class distance and cosine distance of domain centroids in the PACS dataset. As depicted in Table 1, we observe a close relationship between Art and Photo, while the distance between Art and Sketch is substantial. Hence, when Art is designated as the target domain on the central server, local data from Photo can make more contributions than ones from Sketch. These findings support our assumption that some source domains may contain more valuable information for the target domain than other domains. Moreover, our investigation entails an examination of the correlation between domain distance and the effectiveness of domain generalization. As shown in Table 2, we can observe a noteworthy trend: a reduction in domain distance corresponds to generalization performance improvements. The findings demonstrate that various source domains exert disparate influences on the inference performance within the target domain. Additionally, ”overall” means involves harnessing datasets from other domains as training data, thereby facilitating the utilization of valuable complementary information derived from G-Prompt and D-Prompt and improving inference performance within the target domain.

C.2. One client containing multi-domains data.

We investigate the efficacy of DiPrompT when one client contains data from multiple source domains in Figure 3. Although all methods inevitably suffer from performance degradation due to domain shift between mixture domains within a client, DiPrompT exhibits superior performance compared to other state-of-the-art methods. These findings affirm the general ability of DiPrompT to various federated domain generalization tasks.

C.3. The performance on seen domains.

As shown in Table 4, we demonstrate the superior performance of DiPrompT compared to other methods, where prompts are trained and tested within the same domains.

Domains	PACS			
	Art	Cartoon	Photo	Sketch
Art	-	355.89	70.59	1878.98
Cartoon	337.52	-	510.78	1189.07
Photo	28.78	441.94	-	2101.81
Sketch	1761.94	973.19	1941.168	-

Table 1. Distance between different domains in the PACS dataset using features from pre-trained ViT-B/16.

Domains	PACS				
	Art	Cartoon	Photo	Sketch	Overall
Art	-	94.68	94.82	93.93	94.97
Cartoon	95.67	-	94.55	95.10	96.25
Photo	99.52	99.46	-	99.26	99.56
Sketch	82.15	81.97	83.45	-	84.72

Table 2. Performance comparison of our proposed DiPrompT under different domains of PACS dataset.

Methods	PACS				
	Art	Cartoon	Photo	Sketch	Avg
PromptFL	91.24	93.90	99.22	79.51	90.96
FedCLIP	91.65	93.26	99.28	80.80	91.24
DiPrompT	93.46	95.56	99.40	81.08	92.37

Table 3. Analysis for multi-domain data in one client.

It can be observed that all approaches exhibit enhanced performance attributed to the elimination of domain shift between the training and testing phases. Notably, our DiPrompT still outperforms other methods.

C.4. The potential influence of class heterogeneity.

To avoid the interference of class heterogeneity, we conducted experiments under an "IID" data setting. Specifically, when dividing the data of a domain into multiple clients, we initially separated data into distinct categories. Subsequently, we evenly distribute data from each category to different clients, ensuring comprehensive coverage of all categories and the quantities from different categories are close in each client. Table 5 showcases that our DiPrompt alleviates the problem of domain generalization rather than class heterogeneity.

Methods	PACS				
	Art	Cartoon	Photo	Sketch	Avg
PromptFL	95.22	97.23	99.58	89.43	95.26
FedCLIP	95.09	97.14	99.58	89.23	95.36
DiPrompT	96.19	97.40	99.64	89.74	95.74

Table 4. Quantitative analysis on seen domains.

Methods	PACS				
	Art	Cartoon	Photo	Sketch	Avg
PromptFL	94.31	96.37	99.64	85.16	93.87
FedCLIP	94.04	96.05	99.58	84.47	93.53
DiPrompT	95.43	97.10	99.70	87.58	94.95

Table 5. Quantitative analysis on IID data settings.

References

[1] Chen Fang, Ye Xu, and Daniel N Rockmore. 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*, pages 1657–1664, 2013. 1

[2] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 1

[3] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017. 1

Algorithm 1 DiPrompt at training time

```
Require: $K$: number of clients; $H$: number of selected clients in each round; $R$: number of training rounds; $T$: number of local iterations; $\mathcal{D}_i$: $i$-th local data; $M$: number of source domains; $V^G$: G-Prompt, $V^D$; D-Prompts: $V^Q$; Q-Prompt;
Initialize $V^G = \{v_1, ..., v_L\}$ $V^D = \left\{V_1^D, ..., V_M^D\right\}$, $V_m^D = \{v_1, ..., v_L, s_m\}$, $V^Q = \{v_1, ..., v_L\}$
RunServer
for each round $r$ from 0 to $R-1$ do
    Randomly select $\{\mathcal{C}_k\}_{k=1}^H$ from $K$ clients;
    for each $k \in [1, H]$ in parallel do
        $V^{G,r+1}$, $V^{D,r+1} \leftarrow$ RunClient($V^{G,r}$, $V^{D,r}$)
    end for
    $V^{G,r+1} \leftarrow \frac{1}{\sum_{i=1}^H(|\mathcal{D}_i|)} \sum_{i=1}^H(|\mathcal{D}_i|) \cdot V_i^{G,r+1}$        ▷ G-Prompt aggregation
    for each $m$ from 0 to $M$ do
        Update from $V_m^r$ to $V_m^{r+1}$ with Eqn. 4
        Update $\hat{V}_m^{r+1}$ with Eqn. 5
    end for        ▷ D-Prompts aggregation
end for
return $V^{G,R}$, $V^{D,R}$;
RunClient($V^{G,r}$, $V^{D,r}$)
$V_k^{G,(0)} \leftarrow V^{G,r}$; $V^{D,(0)} \leftarrow V^{D,r}$;
for each local iteration $t$ from 0 to $T-1$ do
    $\mathcal{D}^k = \{x_i, y_i\}_{k=1}^{|\mathcal{D}_i|}$
    Update from $V_Q^{(0)}$ to $V_Q^{(1)}$ with Eqn. 7
    Select $m$-th D-Prompt $V_{k,m}^D$ from $V_k^D$ with Eqn. 6
    Update from $V_k^{G,(0)}$ to $V_k^{G,(1)}$ with Eqn. 2
    Update from $V_{k,m}^{D,(0)}$ to $V_{k,m}^{D,(1)}$ with Eqn. 3        ▷ Simultaneous update G-Prompt and D-Prompts
end for
Send $V_k^{G,(T)} \leftarrow V^{G,r}$; $V^{D,(T)}$ to the server and set $V_Q^{(0)} = V_Q^{(T)}$ in local client
```