Global and Local Prompts Cooperation via Optimal Transport for Federated Learning
Supplementary Materials

Supplementary organization:

A. Method Details

A.1. Efficient Scaling Dykstra's Algorithm

As introduced in [6], Problem (8) can be solved by a fast implementation of Dykstra's Algorithm by only performing matrix-vector multiplications, which is very similar to the widely-used and efficient Sinkhorn Algorithm [13]. The details of this algorithm are shown in Algorithm 1.

Algorithm 1: Efficient Scaling Dykstra's algorithm

```
Input: Cost matrix $C$, marginal constraints vectors $\alpha$ and $\beta$, entropic regularization weight $\lambda$.
Initialize: $Q \leftarrow e^{-C/\lambda}$, $v^{(0)} \leftarrow \mathbb{1}_\beta$, $\Delta_v = \infty$, $\epsilon = 0.001$;
Compute: $Q_\alpha \leftarrow \frac{Q}{\text{diag}(\alpha)\mathbb{1}_{|\alpha|\times|\beta|}}$, $Q_\beta^\top \leftarrow \frac{Q^\top}{\text{diag}(\beta)\mathbb{1}_{|\beta|\times|\alpha|}}$;
for $n = 1, 2, 3, \cdots$ do
    $u^{(n)} \leftarrow \min\left(\frac{\mathbb{1}_{|\alpha|}}{Q_\alpha v^{(n-1)}}, \mathbb{1}_{|\alpha|}\right)$ ;
    $v^{(n)} \leftarrow \frac{\mathbb{1}_{|\beta|}}{Q_\beta^\top u^{(n)}}$;
    $\Delta_v = |v^{(n)} - v^{(n-1)}|$;
    if $\Delta_v < \epsilon$ then
        break
    end
end
return $\text{diag}(u^{(n)}) Q \text{diag}(v^{(n)})$
```

A.2. Training Process

Here, we provide detailed descriptions of the algorithm for our FedOTP, as shown in Algorithms 2. For each communication round t, the selected clients perform local training by training global and local prompts $P_i^t = [P_g^t, p_{l,i}^t]$ through unbalanced OT at the same time. Then the updated global prompts $P_{g,i}^t$ are sent to the server for aggregation.

Algorithm 2: FedOTP: Federated Prompts Cooperation via Optimal Transport

```
Input: Communication rounds $T$, local epochs $R$, client number $N$, local dataset $D_i$, sample numbers $m_i$,
       pre-trained CLIP model $g(\cdot)$ and $h(\cdot)$, class number $K$, learning rate $\eta$, temperature of Softmax $\tau$.
Initialize parameters $P_i^0 = [P_g^0, P_{l,i}^0]$;
for each communication round $t \in \{1, \cdots, T\}$ do
    Sample a client set $C^t \subset \{1, \cdots, N\}$ ;
    for each client $i \in C^t$ do
        Initialize $P_i^{t,0} = [P_g^{t-1}, P_{l,i}^{t-1}]$;
        for each local epoch $r \in \{1, \cdots, R\}$ do
            Sample a mini-batch $B_i \in D_i$;
            Obtain a visual feature map $G_m$ with the visual encoder $g(x)(x \in B_i)$;
            Obtain textual features $H_k$ of each class with the textual encoder $\{h(P_{i,k}^{t,r-1})\}|_{k=1}^{K}$;
            Calculate the cost matrix $C_k = 1 - G_m^\top H_k$ of each class;
            Solve Problem (8) through Algorithm 1 and obtain Wasserstein distance $d_{C,k} = \langle T_k^*, C_k \rangle$;
            Calculate the classification probability $q(y = k|\mathbf{x}) = \frac{\exp((1-d_{C,k})/\tau)}{\sum_{c=1}^{K} \exp((1-d_{C,c})/\tau)}$;
            Update the parameters of prompts $P_i^{t,r} \leftarrow P_i^{t,r-1} - \eta \nabla \mathcal{L}_{\mathcal{D}_i}(P_i^{t,r-1})$;
        end
    end
    Aggregate the global prompt $P_g^t = \sum_{i \in C_t} \frac{m_i}{\sum_{j \in C_t} m_j} P_{g,i}^{t,R}$;
end
return $P_i = [P_g, P_{l,i}]$
```

B. Experimental Details

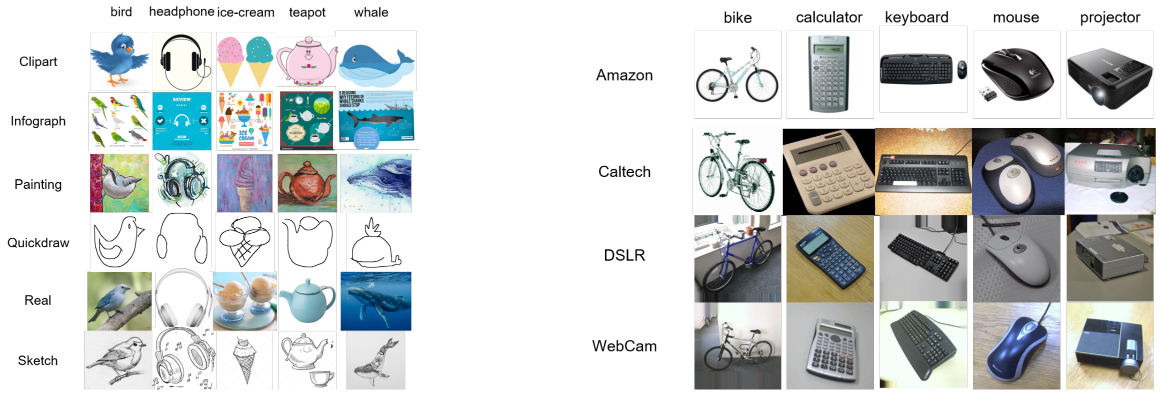
B.1. Details of Dataset Setup

We select nine representative visual classification datasets as our benchmark. The detailed statistics of each dataset are shown in Table A1, including the original tasks, the number of classes, the size of training and testing samples, and the number of domains. As for datasets with multiple domains, Office-Caltech10 is a standard benchmark dataset consisting of four

Table A1. The detailed statistics of datasets used in experiments.

Dataset	Task	Classes	Training Size	Testing Size	Domains
Caltech101 [20]	Object recognition	100	4,128	2,465	1
Flowers102 [54]	Fine-grained flowers recognition	102	4,093	2,463	1
OxfordPets [56]	Fine-grained pets recognition	37	2,944	3,669	1
Food101 [3]	Fine-grained food recognition	101	50,500	30,300	1
DTD [11]	Texture recognition	47	2,820	1,692	1
CIFAR-10 [36]	Image Classification	10	50,000	10,000	1
CIFAR-100 [36]	Image Classification	100	50,000	10,000	1
DomainNet [58]	Image recognition	10	18278	4573	6
Office-Caltech10 [25]	Image recognition	10	2025	508	4

domains, namely Amazon, Caltech, DSLR, and WebCam, which are acquired using different camera devices or in different real environments with various backgrounds. DomainNet is a large-scale dataset consisting of six domains, namely Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. We selected 10 classes from each of these two datasets for training. Some examples of raw instances of these two datasets can be found in Figure A1. For a clearer illustration, we visualize the three Non-IID settings employed in our paper in Figure A2.



(a) DomainNet

(b) Office-Caltech10

Figure A1. Examples of raw instances from two datasets with multiple domains: DomainNet (left) and Office-Caltech10 (right). We present five classes for each dataset to show the feature shift across their sub-datasets.

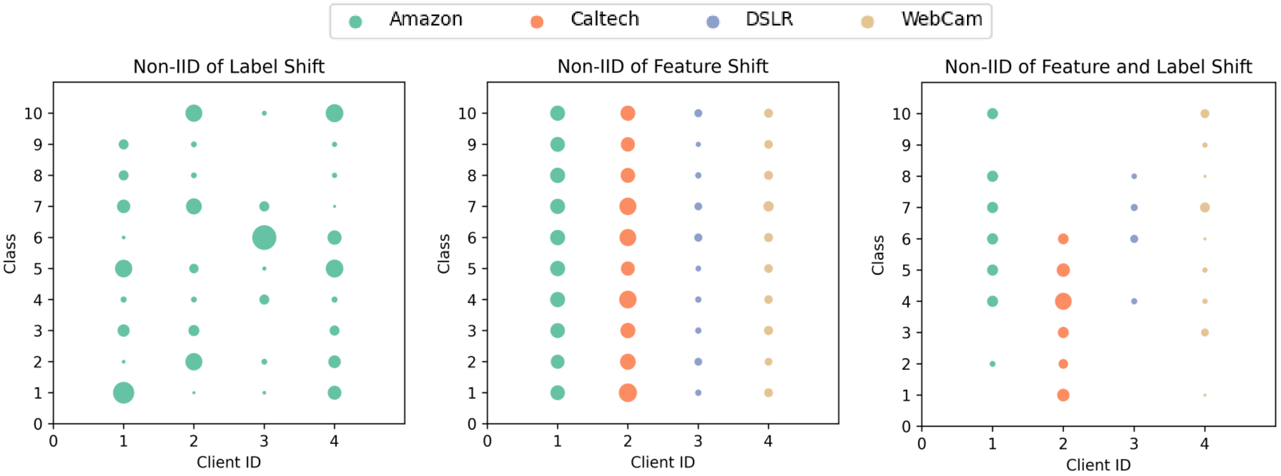


Figure A2. Visualization of three Non-IID settings on the Office-Caltech10 dataset. Each dot represents a set of samples within specific classes assigned to a client, with the dot size indicating the number of samples. The feature shifts are denoted by different colors.

B.2. Implementation Details

All input images across datasets are resized to 224×224 pixels and further divided into 14×14 patches with a dimension of 768. Regarding the hyperparameters for solving OT, we set the entropic regularization weight in Problem Eq. (8) as $\lambda = 0.1$ for all datasets. The maximum iteration number n for Algorithm 1 is set to 100, and we implement early stopping when the absolute update value Δ_v is less than 0.001. For the setting of learnable prompts, the length of prompt vectors s is set to 16 with a dimension of 512, "end" token position, and "random" initialization. Batch sizes are set to 32 for training and 100 for testing. All experiments are conducted with Pytorch [57] on NVIDIA A40 GPUs.

C. Additional Experiments Results

C.1. Model Evaluation on Feature Shifts

In Table A2, we compared the performance on Office-Caltech10 and DomainNet datasets under the presence of feature shift, where each client is assigned data from distinct domains while sharing the same label distribution. Our method achieved the highest average accuracies 99.16% and 94.55% on Office-Caltech10 and DomainNet, respectively.

Table A2. Experimental results on Office-Caltech10 and DomainNet datasets with feature shift.

Datasets	Office-Caltech10					DomainNet						
Domains	A	C	D	W	Avg.	C	I	P	Q	R	S	Avg.
Local Training												
Zero-Shot CLIP [60]	19.3	18.2	21.9	18.6	19.50	49.92	47.15	53.63	31.3	48.4	50.18	46.76
CoOp [78]	96.38	97.24	100	98.31	97.98	98.32	83.01	98.18	82.37	98.21	97.70	92.95
Prompt-based Federated Learning												
PromptFL [27]	96.41	96.39	96.90	100	97.42	98.23	79.91	97.89	66.52	96.83	97.31	89.45
PromptFL+FedProx [42]	97.93	97.21	96.89	100	98.01	98.45	72.32	96.00	63.51	96.08	98.04	87.40
FedOTP (Ours)	**97.92**	**98.68**	**100**	**100**	**99.16**	**98.93**	**84.52**	**98.89**	**87.87**	**98.64**	**98.42**	**94.55**

C.2. Model Evaluation on Feature & Label Shifts

In this set of experiments, we investigated scenarios involving both feature shifts and label shifts by dividing data within a domain into three clients based on the Dirichlet distribution with $\alpha = 0.1$ for the Office-Caltech10 dataset. We calculated the mean and standard deviation of clients in the same domain, and the outcomes are presented in Table A3. Comparing these results with those in Table A2, we can observe that the introduction of label shift leads to a performance decrease across all methods, with federated learning methods employing a shared prompt experiencing the most significant decline. In spite of this, our FedOTP consistently achieves the highest average accuracy, demonstrating its capability to utilize both global and local prompts to capture general domain-invariant and specific domain-specific knowledge for effective adaptation to extreme data heterogeneity.

Table A3. Experimental results on Office-Caltech10 dataset with feature & label shifts.

Datasets	Office-Caltech10				
Domains	Amazon	Caltech	DSLR	Webcam	Avg.
Local Training					
Zero-Shot CLIP [60]	8.45±1.49	6.01±4.25	12.92±9.15	6.48±4.82	8.46±6.26
CoOp [78]	**25.59±6.60**	**36.23±16.97**	30.30±5.18	22.56±5.46	28.67±11.13
Prompt-based Federated Learning					
PromptFL [27]	10.92±4.36	10.37±12.63	15.45±15.34	15.90±17.33	13.16±13.60
PromptFL+FedProx [42]	11.05±4.70	12.04±10.65	19.70±21.74	12.56±12.72	13.84±14.29
FedOTP (Ours)	23.59±4.74	31.64±5.25	**43.94±5.67**	**35.51±9.19**	**33.67±9.76**

C.3. Effect of Parameter γ in Unbalanced OT

In this subsection, we delved into the effect of parameter γ in unbalanced OT, which regulates the mapping size of prompts on the feature map. We conducted experiments on the Pathological Non-IID setting across four datasets with varying numbers of shots and different values of the parameter γ in our FedOTP. Specifically, we set $R = 5$ and $T = 10$ for these experiments. The results presented in Table A4 reveal a notable trend: as the parameter γ decreases, the overall performance initially increases and subsequently decreases. Interestingly, the majority of optimal results are observed at $\gamma = 0.8$ or $\gamma = 0.7$. This observation implies that the optimal alignment between global and local prompts and the feature map is achieved when the mapping size of prompts on the feature map is around $70\% - 80\%$. Consequently, we adopt $\gamma = 0.8$ in other experiments.

Table A4. Quantitative comparisons on the Pathological Non-IID setting across varying numbers of shots with different parameter γ in our FedOTP over 10 clients.

Dataset	shot number	1	0.9	0.8	0.7	0.6	0.5
DTD	1 shot	74.22±0.75	73.72±0.79	75.75±0.64	72.81±0.42	**77.36±0.98**	77.22±1.46
	2 shots	81.89±0.76	84.03±0.57	84.64±0.29	**85.50±0.35**	80.39±0.24	82.47±0.40
	4 shots	85.06±0.91	85.75±0.63	86.69±0.61	**87.67±0.70**	86.58±0.51	85.86±0.44
	8 shots	88.64±0.31	88.22±0.30	89.77±0.24	**90.25±0.74**	89.17±0.53	89.67±0.51
	16 shots	90.51±0.11	91.02±0.48	**91.31±0.59**	90.94±0.25	89.97±0.34	90.33±0.51
Caltech101	1 shot	89.68±1.19	92.13±0.58	**92.54±0.71**	91.53±0.55	90.10±1.74	90.67±0.44
	2 shots	**95.05±0.49**	93.89±0.35	94.45±0.32	94.37±0.43	93.89±0.65	94.68±0.92
	4 shots	96.02±0.36	96.64±0.41	**97.02±0.36**	96.68±0.46	96.38±0.42	96.66±0.37
	8 shots	96.74±0.21	96.79±0.24	96.91±0.16	96.95±0.26	97.22±0.33	**97.34±0.18**
	16 shots	97.72±0.14	97.69±0.17	97.39±0.11	97.58±0.23	97.74±0.19	**97.83±0.18**
Flowers102	1 shot	86.68±1.93	85.77±0.74	87.42±0.92	88.43±0.90	**89.14±1.18**	85.56±1.21
	2 shots	93.09±1.26	**93.96±0.48**	93.31±0.55	93.13±0.26	93.70±0.49	93.56±0.86
	4 shots	95.46±0.55	**96.23±0.44**	95.51±0.30	95.89±0.50	96.17±0.47	96.16±0.34
	8 shots	97.53±0.24	97.49±0.19	**98.23±0.32**	98.11±0.27	97.24±0.28	97.40±0.64
	16 shots	98.86±0.15	98.30±0.55	**99.11±0.11**	98.88±0.17	99.03±0.12	98.93±0.18
OxfordPets	1 shot	95.82±1.16	94.26±0.38	96.18±0.71	**96.37±0.79**	94.21±0.53	95.97±0.42
	2 shots	**97.73±0.57**	96.12±0.32	97.50±1.02	97.49±0.45	97.60±0.40	97.27±0.19
	4 shots	98.11±1.15	98.46±0.64	**98.82±0.11**	98.51±0.10	98.52±0.25	98.43±0.28
	8 shots	98.73±0.27	**99.02±0.38**	98.71±0.16	98.74±0.18	98.54±0.22	98.63±0.13
	16 shots	99.04±0.16	98.82±0.25	**99.27±0.23**	99.21±0.19	99.04±0.27	98.81±0.21

C.4. Effect of Heterogeneity in Label Distribution

In addressing the core challenge of data heterogeneity in personalized federated learning, FedOTP consistently outperforms benchmark methods across various settings. Now, we investigated the effect of heterogeneity in label distribution by considering a range of α values of Dirichlet distribution, specifically $\alpha \in \{0.1, 0.3, 0.5, 1, 5, 10\}$ for CIFAR-100 datasets. It's worth noting that a smaller α implies a higher degree of data heterogeneity in these experiments. The results presented in Table A5 clearly indicate that as the degree of data heterogeneity increases, the performance of federated learning methods with a shared prompt decreases while the performance of CoOp and our FedOTP improves. Among these methods, FedOTP outperforms them in every case and demonstrates remarkable robustness. These findings underscore the effectiveness of FedOTP in overcoming label distribution heterogeneity across a diverse range of scenarios.

C.5. Learning Curves

To assess the convergence of our method, we plotted test accuracy curves with $R = 1$ and $T = 50$ for different methods across four datasets, as illustrated in Figure A3. Compared to other methods, FedOTP exhibits notable characteristics of accelerated convergence and enhanced stability, evident from the smaller fluctuations in test accuracy.

D. Visualization

D.1. Visualizations of Transport Plans

To facilitate a comparative analysis between FedOTP, PromptFL, and Local models, we examined visualizations of similarity between textual features and feature maps in Figure 2. Here, we provided visualization examples showcasing transport plans T associated with global and local prompts of FedOTP across different γ values. We converted each transport plan into colorful heatmaps, resized and overlayed them on the original image. The comparisons between heatmaps of transport plans

Table A5. Quantitative comparisons on CIFAR-100 dataset with different α of the Dirichlet setting.

Dataset	CIFAR-100					
$\#\alpha$	0.1	0.3	0.5	1	5	10
Local Training						
Zero-Shot CLIP [60]	65.22±0.32	64.92±0.53	65.78±0.41	63.93±0.16	64.01±0.27	65.07±0.35
CoOp [78]	62.01±0.29	74.83±0.45	51.72±0.42	47.03±0.37	41.03±0.23	41.37±0.19
Prompt-based Federated Learning						
PromptFL [27]	72.45±0.64	73.67±0.56	74.37±0.18	73.95±0.14	74.68±0.05	74.43±0.08
PromptFL+FedProx [42]	72.57±0.54	71.11±0.91	74.45±0.19	74.19±0.06	74.23±0.09	74.53±0.07
FedOTP (Similarity Averaging)	78.68±0.17	75.70±0.27	75.28±0.12	74.88±0.16	74.48±0.05	74.31±0.39
FedOTP (Classical OT)	79.93±0.19	77.86±0.09	75.76±0.12	75.38±0.08	75.01±0.05	74.73±0.05
FedOTP (Unbalanced OT)	**80.56±0.12**	**78.03±0.08**	**76.75±0.10**	**76.17±0.13**	**75.75±0.03**	**75.52±0.06**

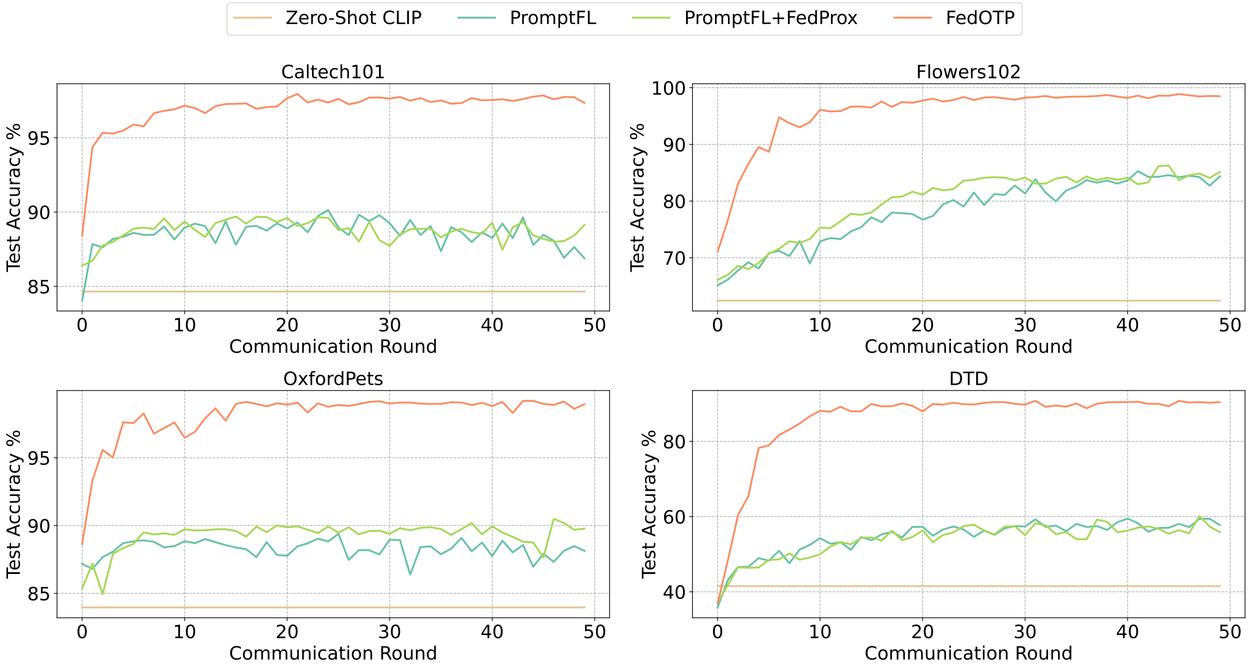


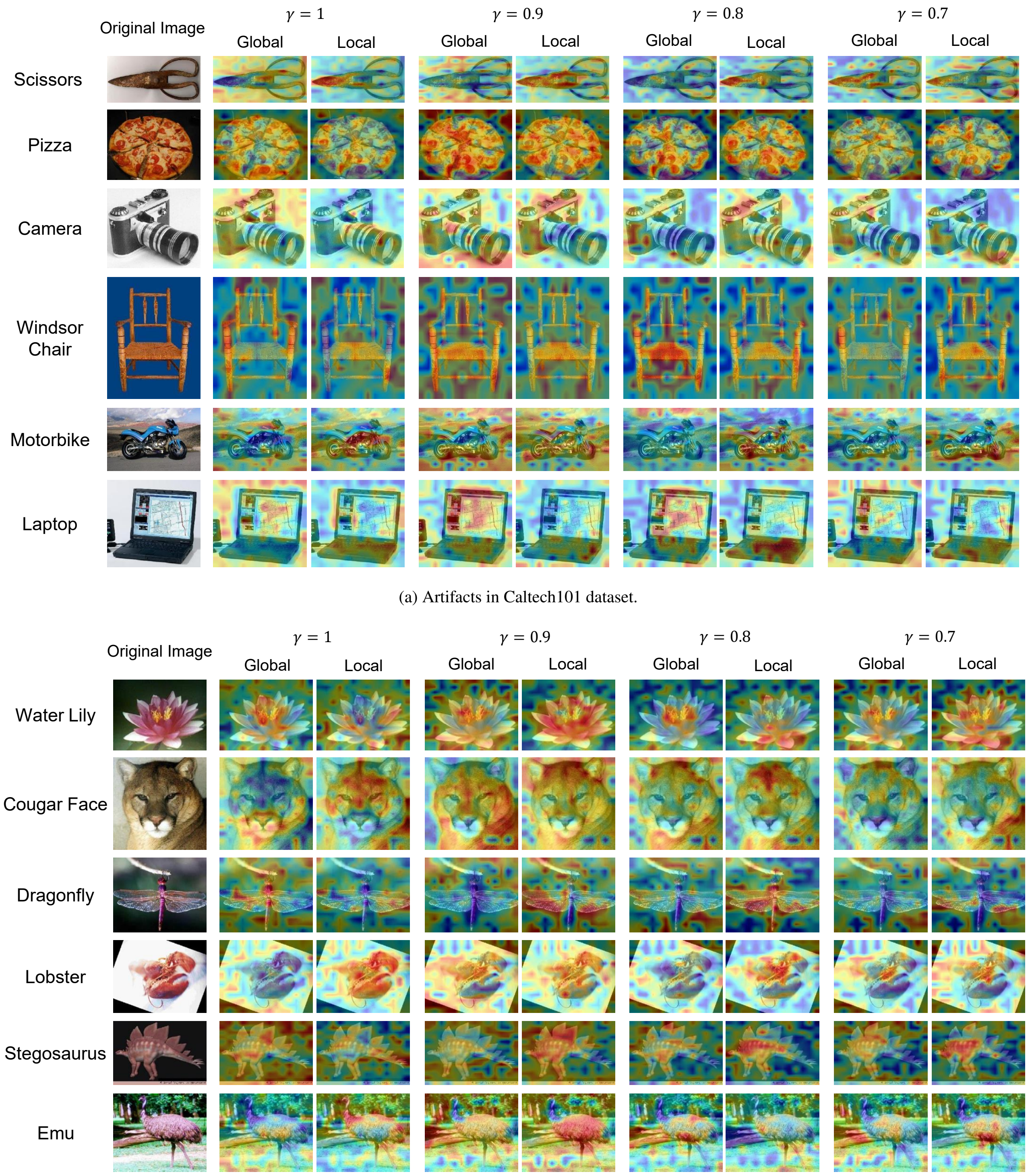
Figure A3. Accuracy curves and convergence behavior of FedOTP and other baselines on four datasets over 10 clients.

with different γ in Caltech101 dataset is presented in Figure A4. Upon observation, we noted that when $\gamma = 1$, the transport plans related to global and local prompts exhibit complementary, as each image patch is assigned to prompts due to the equality constraints of OT. This may result in the integration of objects and backgrounds on the global part, observable in classes like "Camera" and "Laptop". In contrast, as γ decreases, prompts focus on a smaller range of patches and primarily center on the patches of main objects rather than backgrounds, further supporting the claim that FedOTP can effectively regulate the mapping size of prompts on the feature map.

usiv

D.2. T-SNE Projection of Prompts.

To examine how the learned prompts form a meaningful representation across the client space, we employed the t-SNE algorithm [71] to project prompts onto a $2D$ plane. Following [39, 62], we divided CIFAR-100 dataset into 100 clients. In detail, each coarse label was assigned to five clients, and the corresponding fine labels were uniformly distributed among those



(a) Artifacts in Caltech101 dataset.

(b) Organisms in Caltech101 dataset.

Figure A4. Heatmaps of transport plans related to global and local prompts of FedOTP with different γ in Caltech101 dataset. "Global" denotes the transport plans related to global prompts and "Local" refers to local prompts.

selected clients. After training these clients with FedOTP, we visualized their local prompts obtained from local training, and we used different colors to represent various coarse labels. As shown in Figure A5, local prompts from clients with the same coarse label are clustered together and positioned far from those with different coarse labels. These results further illustrate that in FedOTP, the acquired local prompts are tailored to capture client-specific category characteristics.

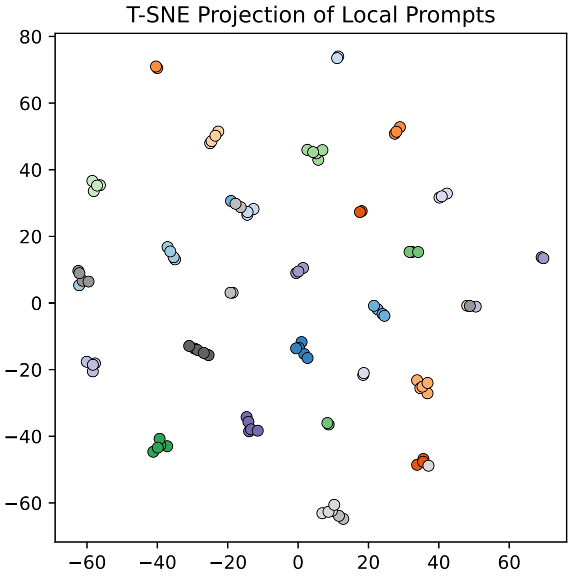


Figure A5. T-SNE projection of local prompts from FedOTP in CIFAR-100 dataset.

E. Generalization Bound

E.1. Key Lemmas

Lemma 1 (McDiarmid's Inequality [53]) *Let $X_1, \cdots, X_n$ be independent random variables, where X_i has range $\mathcal{X}_i$. Let $g : \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathbb{R}$ be any function with the $(a_1, \cdots, a_n)$−bounded difference property: for every $i = 1, \cdots, n$ and $x_1, \cdots, x_n, x'_i \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$, we have*

$$\sup_{x_i \in \mathcal{X}_i} |g(x_1, \cdots, x_i, \cdots, x_n) - g(x_1, \cdots, x'_i, \cdots, x_n)| \leq a_i. \tag{13}$$

Then for any $\varepsilon > 0$,

$$\mathbb{P}[g(X_1, \cdots, X_n) - \mathbb{E}[g(X_1, \cdots, X_n)] \geq \varepsilon] \leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^{n} a_i^2}\right). \tag{14}$$

Lemma 2 (Rademacher Complexity [53]) *Given a space $\mathcal{B}$ and a fixed distribution D_B, let $\{b_1, \cdots, b_m\}$ be a set of examples drawn i.i.d. from D_B. Let $\mathcal{F}$ be a class of functions $f : B \to \mathbb{R}$, and the Rademacher Complexity of $\mathcal{F}$ is defined as follows:*

$$\Re_{D_B}(\mathcal{B}) = \frac{1}{m}\mathbb{E}_\sigma\left[\sup_{b\in B}\sum_{i=1}^{m}\sigma_i b_i\right]. \tag{15}$$

where $\sigma_1, \cdots, \sigma_m$ are independent random variables uniformly chosen from $\{-1, 1\}$.

E.2. Proof of Theorem 1

Proof: For the left side of Theorem 1, we have

$$\begin{aligned}
&\left|\sum_{i=1}^{N}\frac{m_i}{M}\left(\mathcal{L}_{\hat{\mathcal{D}}_i}(\hat{P}_g,\hat{P}_{l,i})-\mathcal{L}_{\mathcal{D}_i}(P_g^*,P_{l,i}^*)\right)\right|\\
&=\left|\sum_{i=1}^{N}\frac{m_i}{M}\Big(\mathcal{L}_{\hat{\mathcal{D}}_i}(\hat{P}_g,\hat{P}_{l,i})-\mathcal{L}_{\hat{\mathcal{D}}_i}(P_g^*,P_{l,i}^*)+\mathcal{L}_{\hat{\mathcal{D}}_i}(P_g^*,P_{l,i}^*)-\mathcal{L}_{\mathcal{D}_i}(P_g^*,P_{l,i}^*)\Big)\right| \qquad (16)\\
&\le\left|\sum_{i=1}^{N}\frac{m_i}{M}\left(\mathcal{L}_{\hat{\mathcal{D}}_i}(\hat{P}_g,\hat{P}_{l,i})-\mathcal{L}_{\hat{\mathcal{D}}_i}(P_g^*,P_{l,i}^*)\right)\right|+\left|\sum_{i=1}^{N}\frac{m_i}{M}\left(\mathcal{L}_{\hat{\mathcal{D}}_i}(P_g^*,P_{l,i}^*)-\mathcal{L}_{\mathcal{D}_i}(P_g^*,P_{l,i}^*)\right)\right|.
\end{aligned}$$

The objective function is partitioned into two components, and we will bound each of them independently. Concerning the first part in Eq. (16), assuming Assumptions 1 and 2 hold, we obtain

$$\begin{aligned}
&\left|\sum_{i=1}^{N}\frac{m_i}{M}\left(\mathcal{L}_{\hat{\mathcal{D}}_i}(\hat{P}_g,\hat{P}_{l,i})-\mathcal{L}_{\hat{\mathcal{D}}_i}(P_g^*,P_{l,i}^*)\right)\right|\\
&\le\left|\sum_{i=1}^{N}\frac{m_i}{M}\Big(\mathcal{L}_{\hat{\mathcal{D}}_i}(\hat{P}_g,\hat{P}_{l,i})-\mathcal{L}_{\hat{\mathcal{D}}_i}(\hat{P}_g,P_{l,i}^*)+\mathcal{L}_{\hat{\mathcal{D}}_i}(\hat{P}_g,P_{l,i}^*)-\mathcal{L}_{\hat{\mathcal{D}}_i}(P_g^*,P_{l,i}^*)\Big)\right|\\
&\le\left|\sum_{i=1}^{N}\frac{m_i}{M}\Big(\mathcal{L}_{\hat{\mathcal{D}}_i}(\hat{P}_g,\hat{P}_{l,i})-\mathcal{L}_{\hat{\mathcal{D}}_i}(\hat{P}_g,P_{l,i}^*)\Big)\right|+\left|\sum_{i=1}^{N}\frac{m_i}{M}\Big(\mathcal{L}_{\hat{\mathcal{D}}_i}(\hat{P}_g,P_{l,i}^*)-\mathcal{L}_{\hat{\mathcal{D}}_i}(P_g^*,P_{l,i}^*)\Big)\right|\\
&\le\sum_{i=1}^{N}\frac{m_i}{M}\mathbb{E}_{(x_i^j,y_i^j)\in D_i}\left|\ell(f(\hat{P}_g,\hat{P}_{l,i};x_i^j),y_i^j)-\ell(f(\hat{P}_g,P_{l,i}^*;x_i^j),y_i^j)\right|\\
&+\sum_{i=1}^{N}\frac{m_i}{M}\mathbb{E}_{(x_i^j,y_i^j)\in D_i}\left|\ell(f(\hat{P}_g,P_{l,i}^*;x_i^j),y_i^j)-\ell(f(P_g^*,P_{l,i}^*;x_i^j),y_i^j)\right| \qquad (17)\\
&\le\sum_{i=1}^{N}\frac{m_i}{M}L\Big(\|f(\hat{P}_g,\hat{P}_{l,i})-f(\hat{P}_g,P_{l,i}^*)\|+\|f(\hat{P}_g,P_{l,i}^*)-f(P_g^*,P_{l,i}^*)\|\Big)\\
&\le\sum_{i=1}^{N}\frac{m_i}{M}\Big(LL_g\|\hat{P}_{l,i}-P_{l,i}^*\|+LL_{l,i}\|\hat{P}_g-P_g^*\|\Big)\\
&\le LL_gA_g+L\sum_{i=1}^{N}\frac{m_i}{M}L_{l,i}A_{l,i}\le LL_gA_g+L\Big(\sum_{i=1}^{N}\frac{m_i}{M}\Big)\Big(\sum_{i=1}^{N}L_{l,i}A_{l,i}\Big)\\
&\le LL_gA_g+L\sqrt{\Big(\sum_{i=1}^{N}L_{l,i}^2\Big)\Big(\sum_{i=1}^{N}A_{l,i}^2\Big)}.
\end{aligned}$$

For the second part in Eq. (16), replacing $g(\cdot)$ with $\sum_{i=1}^{N}\frac{m_i}{M}\left(\mathcal{L}_{\hat{\mathcal{D}}_i}(P_g^*,P_{l,i}^*)-\mathcal{L}_{\mathcal{D}_i}(P_g^*,P_{l,i}^*)\right)$ in Lemma 1, and setting $\delta=\exp\left(-2\varepsilon^2/\sum_{i=1}^{n}a_i^2\right)$, with a probability at least $1-\delta$, the following inequality holds,

$$\begin{aligned}
&\left|\sum_{i=1}^{N}\frac{m_i}{M}\left(\mathcal{L}_{\hat{\mathcal{D}}_i}(P_g^*,P_{l,i}^*)-\mathcal{L}_{\mathcal{D}_i}(P_g^*,P_{l,i}^*)\right)\right| \qquad (18)\\
&\le\mathbb{E}\left[\sum_{i=1}^{N}\frac{m_i}{M}\left(\mathcal{L}_{\hat{\mathcal{D}}_i}(P_g^*,P_{l,i}^*)-\mathcal{L}_{\mathcal{D}_i}(P_g^*,P_{l,i}^*)\right)\right]+\sqrt{\frac{M}{2}log\frac{N}{\delta}}.
\end{aligned}$$

Utilizing Lemma 2 and the results in [50], we can get

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{N}\frac{m_i}{M}\left(\mathcal{L}_{\hat{\mathcal{D}}_i}(P_g^*, P_{l,i}^*) - \mathcal{L}_{\mathcal{D}_i}(P_g^*, P_{l,i}^*)\right)\right] \leq \sum_{i=1}^{N}\frac{m_i}{M}\Re_{\mathcal{D}_i}(\mathcal{H}) \\
&\leq \sum_{i=1}^{N}\frac{m_i}{M}\sqrt{\frac{dN}{m_i}log\frac{em_i}{d}} \leq \sum_{i=1}^{N}\frac{m_i}{M}\sqrt{\frac{dN}{m_i}log\frac{eM}{d}} \leq \sqrt{\frac{dN}{M}log\frac{eM}{d}}.
\end{aligned} \tag{19}$$

Combining the results in Eq. (18) and Eq. (19), we can get

$$\left|\sum_{i=1}^{N}\frac{m_i}{M}\left(\mathcal{L}_{\hat{\mathcal{D}}_i}(P_g^*, P_{l,i}^*) - \mathcal{L}_{\mathcal{D}_i}(P_g^*, P_{l,i}^*)\right)\right| \leq \sqrt{\frac{M}{2}log\frac{N}{\delta}} + \sqrt{\frac{dN}{M}log\frac{eM}{d}}. \tag{20}$$

Summarizing the results above, we can obtain

$$\begin{aligned}
\left|\sum_{i=1}^{N}\frac{m_i}{M}\left(\mathcal{L}_{\hat{\mathcal{D}}_i}(\hat{P}_g, \hat{P}_{l,i}) - \mathcal{L}_{\mathcal{D}_i}(P_g^*, P_{l,i}^*)\right)\right| &\leq \sqrt{\frac{M}{2}log\frac{N}{\delta}} + \sqrt{\frac{dN}{M}log\frac{eM}{d}} + LL_gA_g + L\sqrt{\left(\sum_{i=1}^{N}L_{l,i}^2\right)\left(\sum_{i=1}^{N}A_{l,i}^2\right)} \\
&= \sqrt{\frac{M}{2}log\frac{N}{\delta}} + \sqrt{\frac{dN}{M}log\frac{eM}{d}} + L(L_gA_g + L_lA_l),
\end{aligned} \tag{21}$$

where we denote $L_l = \sqrt{\sum_{i=1}^{N} L_{l,i}^2}$ and $A_l = \sqrt{\sum_{i=1}^{N} A_{l,i}^2}$ for simplicity. ■