

## B Training Details

### B.1 Models

To align with previous works [21], we adapt the pretrained models from CLIP [9] to FPL. Specifically, we use a ResNet-50 model as the image feature encoder and a Transformer [69] model with 63 million parameters and 8 attention heads as the textual feature encoder. We freeze the weights of image and text encoders and only tune the learnable prompt for *text input*.

### B.2 Data Heterogeneity

#### B.2.1 Feature-Shift Data Heterogeneity

In a FL system, the participating clients may collect data from distinct domains. This introduces the training-time feature shift data heterogeneity that could hinder the generalization of obtained models. In light of this, we evaluate the resilience of FPL algorithms under such data heterogeneity in addition to label distribution data heterogeneity. We use the **DomainNet** [70] dataset consisting of six domains, each representing a distinct visual domain such as **Clipart**, **Painting**, **Real**, **Quickdraw**, **Infograph**, and **Sketch**. Figure 4 exemplifies the images sampled from these domains. For each domain, we assign the training data to two clients, resulting in 12 clients with each client only possessing training data from a single domain.



Figure 4: An illustration of images sampled from DomainNet [70]<sup>3</sup>.

#### B.2.2 Class-Shift Data Heterogeneity

We adopt two **data partition strategies** for local clients, *i.e.*, the IID and non-IID, according to FL scenarios or metrics we evaluated:

- The **IID** dataset partition is only applied to evaluate the fewshot generalization performance of global models. Concretely, we assign each client equal training samples of each class based on the number of shots to evaluate the performance under limited training data.
- The **non-IID** dataset partition is applied to evaluate the performance of methods under label distribution data heterogeneity. We take the partition strategy from [62] to simulate heterogeneous data partition based on a Dirichlet distribution controlled by a concentration parameter  $\alpha$ . A smaller  $\alpha$  indicates more aggressive data heterogeneity, while an infinitely large  $\alpha$  degenerates to the *i.i.d.* data partition. We set  $\alpha = 0.1$  by default in all experiments unless otherwise stated. To partition the original training data for multiple clients, we first subsample a smaller balanced dataset from the original training set, and then apply the Dirichlet data partition. For all experiments, we subsample 8 images for each class to construct the dataset for partition except for few-shot experiments mentioned above. Besides, we set 16 images for each class for partial participation scenarios.

<sup>3</sup><https://ai.bu.edu/M3SDA/>

Notably, the experimental setting on evaluating DomainNet introduces additional feature shift data heterogeneity raised by domain-specific features on each client. This can be considered as an extension on the non-*i.i.d.* data partition with domain heterogeneity.

### B.3 Methods

We consider the following methods as baselines of FPL. For more comprehensive comparison, we also include centralized methods, such as CoCoOp, PLOT, ProDA, ProGrad, Prompt-SRC and KgCoOp. All centralized methods are implemented for local client training and combined with FedAvg [4] for global aggregation. The evaluation helps to understand the properties of existing PL methods under a broad range of evaluation metrics for federated training. This also conveys insights of the appropriate application scenarios of each method under federation.

Below we make a brief introduction of FL methods:

- **PromptFL** [21] is a simple yet effective FPL method that can be viewed as a federated variant of CoOp [11] with FedAvg [4] for global aggregation. PromptFL only communicates the shared soft prompts instead of a shared global model as in conventional FL. This drastically reduces the communication cost of FL. We evaluate PromptFL in all experiments as an important baseline.
- **FedOTP**<sup>4</sup> [16] extends upon the PLOT [18] and designs a novel optimization scheme for imbalanced optimal transport. It also proposes to learn both local and global aligned representation for better generalization. The FedOTP is originally designed for personalized FL. To evaluate its performance on generic FL, we make a small alteration to allow all local prompt parameters to be communicated and updated. This allows the evaluation of FedOTP in most FPL scenarios.
- **FedTPG**<sup>5</sup> [60] designs a text-driven prompt generation network, which is conditioned on task-related text input, enabling robust generalization to both seen and unseen classes.. We evaluate FedTPG in most FPL scenarios except for cost-performance trade-off due to its employment of attention modules, which incur significantly larger communication overhead than other methods.
- **FedPGP**<sup>6</sup> [23] strikes a balance between personalization and generalization of FPL via low-rank adaptation and contrastive learning. As it focuses on personalized performance and generalization capability of client models. We evaluate it in personalized PFL and base-to-novel generalization scenarios.
- **PromptFolio**<sup>7</sup> [22] introduces a portfolio consisting of global prompt and local prompt to balance the generalization and personalization, motivated by portfolio optimization. This work also establishes a theoretical analysis framework for FPL based on feature learning theory. We evaluate it in the personalized FPL scenario.
- **DP-FPL**<sup>8</sup> [29] leverages global and local differential privacy to achieve a privacy-preserving personalized FPL approach for multi-modal LLMs. We evaluate it in the personalized FPL scenario.

In addition, we also evaluate a rich set of centralized prompt learning methods by seamlessly adapting them to FPL as a *local training methods* on clients. These methods are comprehensively evaluated in all FPL scenarios with the exception of MaPLe [71], which requires tuning the image and textual prompts in their corresponding Transformer models. Therefore, we only evaluate it on FPL scenarios with the vision Transformer (ViT-B/16) as the image encoder (Appendix C.3). Below are the details of evaluated algorithms:

- **CoCoOp**<sup>9</sup> [12] addresses the base- and novel-classes generalization dilemma of PL by introducing conditional inference. It optimizes an additional meta-net to deliver the im-

<sup>4</sup><https://github.com/HongxiaLee/FedOTP>

<sup>5</sup><https://github.com/boschresearch/FedTPG>

<sup>6</sup><https://github.com/TianyuCui0v0/FedPGP>

<sup>7</sup><https://github.com/PanBikang/PromptFolio>

<sup>8</sup><https://github.com/linhhtran/DP-FPL>

<sup>9</sup><https://github.com/KaiyangZhou/CoOp>

age features for complementing the domain specific information during inference, which alleviates the over-fitting issue of the CoOp [11].

- **PLOT**<sup>[10]</sup> [18] introduces optimal transport to match the image and textual features. This benefits the distribution alignment of cross-modality features to reduce the modality gaps.
- **ProDA**<sup>[11]</sup> [13] proposes a optimization framework to learn a Gaussian distribution over possible prompts rather than relying on a single static prompt. It also prompts the diversity of prompt sets by introducing a orthogonality regularization loss term.
- **ProGrad**<sup>[12]</sup> [19] updates the prompt with aligned gradient (or non-conflicting) to the general knowledge which is achieved by regularizing gradient update with tailored prompts for domain-specific dataset.
- **KgCoOp**<sup>[13]</sup> [15] is a concurrent work that also introduces tailored prompts for each dataset as an anti-overfitting technique for guiding the prompt optimization. This reduces the discrepancy between the textual features produced by learnable prompts and the hand-crafted prompts, enhancing the generalization ability for unseen classes.
- **SRC**<sup>[14]</sup> [17] regularizes the PL with the predictions of the frozen model, multiple prompts over the training trajectory and textual diversity from different prompt templates. It reduces the catastrophic forgetting of generalizable knowledge from the pretrained CLIP models.
- **MaPLc**<sup>[15]</sup> [71] promotes better vision-language alignment on downstream tasks by introducing multi-modal PL. It also employs a coupling function to condition vision prompts on language counterparts, acting as a bridge between two modalities. We evaluate MaPLc on FPL scenarios with the vision Transformer (ViT-B/16) as the image encoder.

To unify the experimental settings for fair and faithful results, we adapt the official public code implementation of these centralized methods into our FL framework with minimal alterations such as renaming their original arguments. For methods that lacked public code, we either re-implement their algorithms or port the unofficial code with careful scrutinizing of the algorithmic details to ensure alignment with original papers. We will continuously include more FPL methods into FLIP.

## B.4 Hyperparameters

### B.4.1 General Hyperparameter Settings

We use the standard SGD optimizer with initial learning rate 0.002, momentum 0.9, and a cosine learning rate decay scheduler to guarantee the sufficient convergence of each method. We set the batch size as 16, global communication rounds 50 and the local training epoch 1. For each run, we use *random* prompt initialization without prompt position augmentation when constructing the entire prompt with prefix, class name and suffix. The input images is resized to  $240 \times 240$  then cropped with size  $224 \times 224$  to match the input image size of CLIP [9] image encoder, followed by random horizontal flipping and normalization. For each experiments, we conduct 3 independent runs with different random seeds and report the mean and standard variance of accuracy on the test set. Notably, Zhou *et al.* [12] splits the base and novel classes based on sorted class names (in alphabetical order). To rigorously benchmark FPL algorithms across different base and novel class splits, we expand the base-to-novel setup with nine random partitions plus the one from [12]. We report the averaged accuracy obtained from these different base and novel dataset partitions.

### B.4.2 Hyperparameters for FPL Algorithms

In addition to above general hyperparameters shared by all evaluated algorithms, we also conducted hyperparameter tuning for each algorithm in our benchmark. The goal was to identify optimal configurations that maximize performance while maintaining consistency across different experimental settings. While enforcing equal hyperparameter tuning budgets across all algorithms is critical for

<sup>10</sup><https://github.com/CHENGY12/PLOT>

<sup>11</sup><https://github.com/bbbdylan/proda> (unofficial).

<sup>12</sup><https://github.com/BeierZhu/Prompt-align>.

<sup>13</sup><https://github.com/htyao89/KgCoOp>.

<sup>14</sup><https://github.com/muzairkhattak/PromptSRC>

<sup>15</sup><https://github.com/muzairkhattak/multimodal-prompt-learning>.

fairness, disparities in the number and complexity of tunable parameters per algorithm introduce challenges in comparative evaluation. Specifically, some algorithms have no tunable hyperparameters other than the general hyperparameters shared by all algorithms, while others may occupy multiple extra hyperparameters. Moreover, the total computational cost grows exponentially *w.r.t.* the total number of hyperparameters of a FPL algorithm, making exhaustive hyperparameter tuning prohibitively expensive. To address this, we adopt a fixed tuning budget for each algorithm with a set of hyperparameter variants similar to those reported in their original papers. We report the averaged results of 3 experimental runs under the best hyperparameter configuration. In Table 9 we detail the hyperparameters explored for each algorithm.

Table 9: The hyperparameters explored for evaluated FPL algorithms.

Algorithms	Hyperparameters	Specification
<b>FedOTP</b>	$\gamma \in [0.7, 0.8, 0.9]$	hyperparameter for unbalanced OT
<b>FedPGP</b>	$\mu \in [0.5, 1, 5]$	tradeoff parameter for the contrastive loss
<b>PromptFolio</b>	$\theta \in [0.1, 0.2, 0.4]$	balancing coefficient
<b>DP-FPL</b>	$C_{th} \in [5, 10, 20]$	clipping threshold
<b><i>f</i>-PLOT</b>	$\lambda \in [0.01, 0.1, 1]$	Entropy regularization hyperparameter for OT
<b><i>f</i>-ProDA</b>	$\lambda \in [0.01, 0.1, 0.5]$	tradeoff parameter for semantic orthogonality
<b><i>f</i>-ProGrad</b>	$\lambda \in [0.4, 0.8, 1]$	tradeoff parameter for gradient regularization
<b><i>f</i>-SRC</b>	$\lambda_1, \lambda_2 \in [(1, 1), (5, 10), (10, 25)]$	balancing coefficients for regularization losses
<b><i>f</i>-KgCoOp</b>	$\lambda \in [1, 4, 8]$	tradeoff parameter of the regularization loss

## C Additional Results

### C.1 Centralized Setting

In Table 10, we report the training accuracy values of PL methods under the centralized setting. *With the initialization of the pretrained models, there is only a slender margin between centralized and federated settings.* We speculate the underlying reason is that rich features from the pretrained models significantly reduce the potential gradient conflict among client updates. This observation holds the promise of closing the gap between centralized and federated training, motivating practical and efficient algorithms that specifically seek out better generalization with pretrained vision-language models.

Table 10: Comparison of training accuracy (%) of PL methods **under the centralized (*i.e.* non-FL) setting**. We report the mean  $\pm$  standard deviation over 3 runs.

Centralized	Caltech	DTD	Aircraft	Food	Cars	Flowers	Pets	UCF
<b>ZS-CLIP</b>	86.0	41.7	16.6	77.9	55.5	65.3	85.7	61.5
<b>CoOp</b>	91.5 $\pm$ 0.8	58.1 $\pm$ 1.0	23.5 $\pm$ 0.8	79.3 $\pm$ 0.3	63.0 $\pm$ 0.2	86.4 $\pm$ 0.1	89.3 $\pm$ 0.5	70.7 $\pm$ 0.3
<b>CoCoOp</b>	91.9 $\pm$ 0.2	57.2 $\pm$ 1.0	19.1 $\pm$ 1.4	79.4 $\pm$ 0.5	62.7 $\pm$ 0.2	79.9 $\pm$ 1.5	88.9 $\pm$ 0.2	68.7 $\pm$ 1.5
<b>PLOT</b>	91.7 $\pm$ 0.3	58.8 $\pm$ 0.4	23.4 $\pm$ 0.8	78.3 $\pm$ 0.1	62.4 $\pm$ 0.6	86.1 $\pm$ 0.2	89.6 $\pm$ 0.3	71.0 $\pm$ 0.1
<b>ProDA</b>	91.8 $\pm$ 0.3	57.0 $\pm$ 0.8	22.8 $\pm$ 0.2	79.0 $\pm$ 0.2	63.6 $\pm$ 0.6	88.6 $\pm$ 0.7	89.0 $\pm$ 0.2	70.9 $\pm$ 0.5
<b>ProGrad</b>	91.2 $\pm$ 0.3	57.8 $\pm$ 1.0	21.7 $\pm$ 1.3	79.4 $\pm$ 0.1	63.3 $\pm$ 0.1	87.9 $\pm$ 0.3	89.1 $\pm$ 1.2	70.1 $\pm$ 0.9
<b>PromptSRC</b>	92.2 $\pm$ 0.1	57.9 $\pm$ 1.6	22.7 $\pm$ 0.2	78.9 $\pm$ 0.1	63.5 $\pm$ 0.2	84.2 $\pm$ 3.2	89.4 $\pm$ 0.1	71.5 $\pm$ 0.3
<b>KgCoOp</b>	91.8 $\pm$ 0.2	58.6 $\pm$ 0.5	23.8 $\pm$ 0.1	79.5 $\pm$ 0.3	64.3 $\pm$ 0.6	84.3 $\pm$ 2.0	89.6 $\pm$ 0.7	71.3 $\pm$ 0.7

### C.2 Cost-performance Trade-offs

Tables 11 and 12 present the communication and performance trade-off by changing the number of prompts or prompt context token length. Figures 5a and 5b present the communication and performance trade-offs by changing the number of prompts and prompt token lengths respectively. First, we note that a direct scaling of the learnable parameters does not necessarily deliver positive improvements. For example, *f*-CoCoOp employs a meta-net to aggregate the conditional image

Table 11: **Trade-offs between accuracy (%) and the number of communicated parameters (in millions) under different number of prompts on Caltech.** Here, we sweep the number of prompts with  $\{1, 2, 4\}$  while keeping the number of prompt tokens fixed at 4.

Number of Prompts	<b>1</b>		<b>2</b>		<b>4</b>		Avg.	#
	Accuracy	Cost	Accuracy	Cost	Accuracy	Cost		
<b>PromptFL</b>	91.5 $\pm$ 0.5	2.05	91.4 $\pm$ 0.4	4.10	91.7 $\pm$ 0.1	8.19	91.6	-
<b>FedOTP</b>	<u>91.8</u> $\pm$ 0.1	4.10	<u>91.8</u> $\pm$ 0.5	8.19	<u>91.9</u> $\pm$ 0.3	16.38	<u>91.8</u>	<b>3</b>
<i>f</i> -CoCoOp	91.7 $\pm$ 0.3	100.93	91.3 $\pm$ 0.3	102.98	91.7 $\pm$ 0.1	107.07	91.5	1
<i>f</i> -PLOT	91.6 $\pm$ 0.3	2.05	91.2 $\pm$ 0.3	4.10	91.4 $\pm$ 0.2	8.19	91.4	1
<i>f</i> -ProDA	91.6 $\pm$ 0.3	4.10	91.1 $\pm$ 0.1	8.19	91.7 $\pm$ 0.1	16.38	91.5	1
<i>f</i> -ProGrad	90.7 $\pm$ 0.2	2.05	91.1 $\pm$ 0.1	4.10	91.4 $\pm$ 0.1	8.19	91.1	0
<i>f</i> -SRC	<b>92.0</b> $\pm$ 0.8	2.05	<b>92.0</b> $\pm$ 0.3	4.10	<b>92.1</b> $\pm$ 0.1	8.19	<b>92.0</b>	<b>3</b>
<i>f</i> -KgCoOp	<u>91.8</u> $\pm$ 0.2	2.05	91.4 $\pm$ 0.2	4.10	<u>91.9</u> $\pm$ 0.2	8.19	91.7	<u>2</u>

Table 12: **Trade-offs between accuracy (%) and the number of communicated parameters (in millions) under different number of prompt tokens on Caltech.** Here, we sweep the number of tokens with  $\{4, 8, 16\}$  while keeping the number of prompts fixed at 1.

Number of Tokens	<b>4</b>		<b>8</b>		<b>16</b>		Avg.	#
	Accuracy	Cost	Accuracy	Cost	Accuracy	Cost		
<b>PromptFL</b>	91.5 $\pm$ 0.5	2.05	91.0 $\pm$ 0.6	4.10	91.7 $\pm$ 0.2	8.19	91.4	-
<b>FedOTP</b>	<u>91.8</u> $\pm$ 0.1	4.10	<u>91.8</u> $\pm$ 0.2	8.19	<u>92.0</u> $\pm$ 0.3	16.38	<u>91.8</u>	<b>3</b>
<i>f</i> -CoCoOp	91.7 $\pm$ 0.3	100.93	91.6 $\pm$ 0.9	102.98	91.8 $\pm$ 0.2	107.07	91.7	<b>3</b>
<i>f</i> -PLOT	91.6 $\pm$ 0.3	2.05	91.4 $\pm$ 0.3	4.10	91.7 $\pm$ 0.1	8.19	91.6	<u>2</u>
<i>f</i> -ProDA	91.6 $\pm$ 0.3	4.10	91.5 $\pm$ 0.6	8.19	91.8 $\pm$ 0.2	16.38	91.6	<b>3</b>
<i>f</i> -ProGrad	90.7 $\pm$ 0.2	2.05	91.0 $\pm$ 0.3	4.10	91.6 $\pm$ 0.1	8.19	91.1	1
<i>f</i> -SRC	<b>92.0</b> $\pm$ 0.8	2.05	<b>92.1</b> $\pm$ 0.3	4.10	<b>92.2</b> $\pm$ 0.1	8.19	<b>92.1</b>	<b>3</b>
<i>f</i> -KgCoOp	<u>91.8</u> $\pm$ 0.2	2.05	91.2 $\pm$ 0.4	4.10	91.4 $\pm$ 0.1	8.19	91.5	<u>2</u>

information, which drastically increases the number of communications. However, this does not translates to accuracy boost over the simple baseline in most experiments. Besides, by comparing the accuracies increments of a single methods with different number of prompt or prompt length, we can conclude that methods with distribution alignment (*f*-FedOTP) or diversity regularization (*f*-ProDA) usually bring in stable improvements when scaling up the prompt parameters. Finally, both approaches for tweaking the prompt parameters yield similar improvements. Indeed, we do not observe clear dominance of them over the other.

### C.3 Evaluation on Transformer Image Encoder

In Tables [13](#) to [15](#) we respectively report the global, personal and base-to-novel accuracy metrics for the ViT-B/16 image encoder, following the evaluation protocols used in Tables [1](#) to [3](#). To sum up, these results show a similar trend on those of ResNet-50, and the ViT-B/16 in most experiments delivers better results than the ResNet-50 image encoder. Notably, *f*-MaPLe manifests clear advantages because of its multi-modal prompt optimization in both image and textual encoders, serving as a strong competitor compared with other federated prompt learning algorithms.

## D Discussion

### D.1 Implementation Details

**Environments** We implement all evaluated methods with PyTorch [\[72\]](#) of version 2.1.0. We try to minimize the number of packages used in our code framework, and setting up the environment only requires minutes. To alleviate computational burden, we apply the automatic mixed precision (AMP)



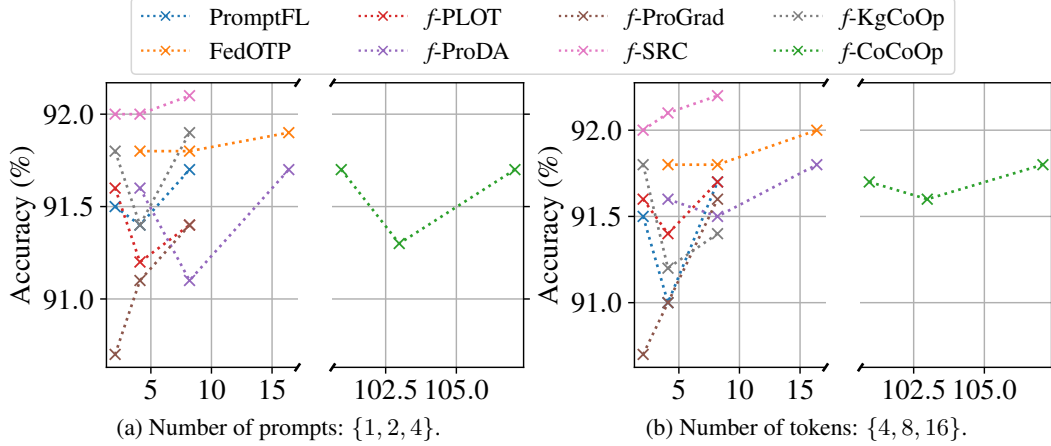


Figure 5: Trade-offs between accuracy (%) and the number of communicated parameters (in millions) on Caltech-101.

Table 13: Comparison of shared global model accuracy  $\alpha_g$  (%) of FPL methods with a ViT-B/16 image encoder. Results are reported in a similar style as Table 1.

Global $\alpha_g$	Caltech	DTD	Aircraft	Food	Cars	Flowers	Pets	UCF	Avg.	#
<b>ZS-CLIP</b>	93.5	45.0	24.3	85.5	65.6	68.0	89.2	67.5	67.3	-
<b>PromptFL</b>	95.5 $\pm$ 0.1	59.6 $\pm$ 0.7	31.2 $\pm$ 0.3	86.8 $\pm$ 0.1	70.2 $\pm$ 1.3	86.2 $\pm$ 1.7	92.4 $\pm$ 0.2	77.5 $\pm$ 0.6	74.9	-
<b>FedOTP</b>	95.6 $\pm$ 0.1	60.6 $\pm$ 0.7	32.4 $\pm$ 0.6	86.9 $\pm$ 0.2	70.6 $\pm$ 0.5	85.6 $\pm$ 0.3	92.2 $\pm$ 0.5	76.6 $\pm$ 0.6	75.1	5
<b>FedTPG</b>	95.4 $\pm$ 0.2	60.2 $\pm$ 0.3	31.4 $\pm$ 0.2	86.8 $\pm$ 0.2	70.2 $\pm$ 0.1	85.7 $\pm$ 0.3	91.8 $\pm$ 0.1	76.2 $\pm$ 0.3	74.8	3
<b>f-CoCoOp</b>	95.6 $\pm$ 0.2	57.7 $\pm$ 1.0	31.0 $\pm$ 0.6	86.6 $\pm$ 0.2	68.5 $\pm$ 0.3	81.6 $\pm$ 0.8	92.4 $\pm$ 0.7	74.5 $\pm$ 0.5	73.5	1
<b>f-PLOT</b>	95.4 $\pm$ 0.2	59.6 $\pm$ 0.9	31.3 $\pm$ 0.4	86.5 $\pm$ 0.1	70.1 $\pm$ 1.3	85.8 $\pm$ 2.2	92.4 $\pm$ 0.2	77.5 $\pm$ 0.3	74.8	3
<b>f-ProDA</b>	95.4 $\pm$ 0.2	58.6 $\pm$ 1.0	31.3 $\pm$ 0.8	86.6 $\pm$ 0.1	70.8 $\pm$ 1.2	84.7 $\pm$ 0.4	92.5 $\pm$ 0.2	77.0 $\pm$ 0.5	74.6	3
<b>f-ProGrad</b>	95.2 $\pm$ 0.1	56.5 $\pm$ 0.4	30.0 $\pm$ 0.4	87.1 $\pm$ 0.1	69.3 $\pm$ 0.1	81.3 $\pm$ 1.4	92.6 $\pm$ 0.3	75.4 $\pm$ 0.5	73.4	2
<b>f-PromptSRC</b>	94.0 $\pm$ 0.5	58.1 $\pm$ 0.3	31.4 $\pm$ 0.2	86.8 $\pm$ 0.1	70.3 $\pm$ 0.2	85.3 $\pm$ 0.6	92.5 $\pm$ 0.2	75.1 $\pm$ 0.1	74.2	3
<b>f-KgCoOp</b>	95.4 $\pm$ 0.1	59.4 $\pm$ 0.6	31.6 $\pm$ 0.5	86.9 $\pm$ 0.1	70.2 $\pm$ 1.0	83.7 $\pm$ 1.5	92.5 $\pm$ 0.4	76.7 $\pm$ 0.2	74.6	4
<b>f-MaPLe</b>	96.2 $\pm$ 0.4	61.2 $\pm$ 0.2	31.8 $\pm$ 0.2	87.6 $\pm$ 0.3	70.3 $\pm$ 0.4	86.8 $\pm$ 0.1	92.8 $\pm$ 0.4	78.1 $\pm$ 0.2	75.6	8

training<sup>16</sup> which leverages the 16-bit floating point format to reduce GPU memory consumption and computation cost. We do not apply the AMP on DP-FPL because it requires operations such as gradient clipping in full precision format.

**Code Framework** To date, there still lacks a comprehensive and reliable evaluation of FPL algorithms for vision tasks. Zhou *et al.* [12] established the first seminal library for *centralized* prompt learning, which is later reused by a line of subsequent works. However, this library is not tailored for *federated learning*. Namely, it poses additional challenges to incorporate various federated algorithms with existing PL techniques in a flexible and scalable way.

To close this gap, we release the first framework with large-scale evaluations to push the frontier of FPL. To harvest the rapid progress from FL and PL literature, we decouple the design of the FL and PL modules, making it easier to integrate the progress from both research fields in a scalable and efficient way. We simplify and unify the interface of data-loading to achieve better adaptation of new datasets and also make it readily available for users to adapt to their customized datasets for new tasks with minimal modification. We plan to actively support more applications beyond the evaluated image classification tasks.

## D.2 Computational Resources

The experiments are conducted on a cluster consisting of multiple servers equipped with NVIDIA A100 graphic cards. We run experiments on servers equipped with the SLURM<sup>17</sup> job scheduler.

<sup>16</sup><https://pytorch.org/docs/stable/amp.html>

<sup>17</sup><https://slurm.schedmd.com/documentation.html>

Table 14: Comparison of personal model accuracy  $\alpha_p$  (%) of FPL methods on various datasets with a ViT-B/16 image encoder.

Personal $\alpha_p$	Caltech	DTD	Aircraft	Food	Cars	Flowers	Pets	UCF	Avg.	#
<b>ZS-CLIP</b>	93.5	45.0	24.3	85.5	65.6	68.0	89.2	67.5	67.3	-
<b>PromptFL</b>	95.7 $\pm$ 0.4	73.3 $\pm$ 0.6	43.6 $\pm$ 0.5	89.1 $\pm$ 0.6	76.7 $\pm$ 1.3	88.5 $\pm$ 1.0	92.8 $\pm$ 1.1	82.5 $\pm$ 0.6	80.3	-
<b>FedOTP</b>	<u>96.2</u> $\pm$ 0.5	<u>75.2</u> $\pm$ 2.2	<b>46.8</b> $\pm$ 1.1	<u>90.1</u> $\pm$ 0.9	<b>77.7</b> $\pm$ 1.8	<b>91.6</b> $\pm$ 0.2	93.1 $\pm$ 0.7	<u>84.2</u> $\pm$ 1.5	<b>81.9</b>	<b>8</b>
<b>FedTPG</b>	95.7 $\pm$ 0.2	73.8 $\pm$ 0.1	45.2 $\pm$ 0.3	88.8 $\pm$ 0.2	76.2 $\pm$ 0.8	89.4 $\pm$ 0.3	92.3 $\pm$ 0.5	83.3 $\pm$ 0.2	80.6	4
<b>FedPGP</b>	95.2 $\pm$ 0.3	73.8 $\pm$ 0.4	45.6 $\pm$ 0.4	88.8 $\pm$ 0.3	76.8 $\pm$ 0.7	87.6 $\pm$ 0.6	92.6 $\pm$ 0.4	82.9 $\pm$ 0.6	80.3	4
<b>PromptFolio</b>	95.8 $\pm$ 0.3	74.2 $\pm$ 0.5	45.4 $\pm$ 0.4	88.9 $\pm$ 0.6	76.5 $\pm$ 0.3	88.4 $\pm$ 0.4	92.6 $\pm$ 0.5	83.8 $\pm$ 0.5	80.7	4
<b>DP-FPL</b>	95.0 $\pm$ 0.3	71.2 $\pm$ 0.5	42.4 $\pm$ 0.2	84.2 $\pm$ 0.8	75.4 $\pm$ 0.5	82.8 $\pm$ 1.6	92.8 $\pm$ 0.4	80.0 $\pm$ 0.6	78.0	0
<i>f-CoCoOp</i>	95.8 $\pm$ 0.5	72.2 $\pm$ 1.7	<u>45.9</u> $\pm$ 0.1	<u>90.1</u> $\pm$ 0.8	75.8 $\pm$ 0.6	88.1 $\pm$ 0.6	<b>93.9</b> $\pm$ 0.7	82.7 $\pm$ 1.1	<u>80.6</u>	<u>5</u>
<i>f-PLOT</i>	95.8 $\pm$ 0.3	71.5 $\pm$ 0.3	44.3 $\pm$ 1.0	88.7 $\pm$ 0.4	76.8 $\pm$ 1.3	88.8 $\pm$ 1.8	92.9 $\pm$ 1.2	82.0 $\pm$ 1.2	80.1	<u>5</u>
<i>f-ProDA</i>	95.8 $\pm$ 0.4	70.3 $\pm$ 1.1	43.3 $\pm$ 1.4	88.7 $\pm$ 0.4	76.9 $\pm$ 1.7	89.1 $\pm$ 2.0	92.9 $\pm$ 0.5	82.0 $\pm$ 0.6	79.9	4
<i>f-ProGrad</i>	95.1 $\pm$ 0.3	70.2 $\pm$ 0.7	44.4 $\pm$ 1.2	88.9 $\pm$ 0.6	75.9 $\pm$ 1.0	87.3 $\pm$ 1.5	92.3 $\pm$ 1.0	81.5 $\pm$ 0.8	79.5	1
<i>f-PromptSRC</i>	94.8 $\pm$ 0.5	71.2 $\pm$ 1.4	44.2 $\pm$ 0.8	88.4 $\pm$ 0.8	76.8 $\pm$ 0.7	88.0 $\pm$ 0.9	92.7 $\pm$ 1.4	82.4 $\pm$ 0.8	79.8	2
<i>f-KgCoOp</i>	95.5 $\pm$ 0.2	71.4 $\pm$ 1.4	40.4 $\pm$ 0.7	88.7 $\pm$ 1.0	75.3 $\pm$ 1.3	88.0 $\pm$ 1.1	92.2 $\pm$ 0.7	81.2 $\pm$ 1.8	79.1	0
<i>f-MaPLe</i>	<b>96.4</b> $\pm$ 0.4	<b>75.8</b> $\pm$ 0.5	45.6 $\pm$ 0.4	<b>90.6</b> $\pm$ 0.4	<u>77.2</u> $\pm$ 0.3	<u>91.2</u> $\pm$ 0.3	<u>93.5</u> $\pm$ 0.6	<b>84.6</b> $\pm$ 0.6	<b>81.9</b>	<b>8</b>

Table 15: Comparison of base and novel class accuracy (%) of FPL methods with a ViT-B/16 image encoder. Evaluation follows Table 3 except the use of a ViT-B/16 image encoder.

	Caltech			Aircraft			Cars			Flowers			Avg.			#
Metric	$\alpha_b$	$\alpha_n$	$\alpha_h$	$\alpha_b$	$\alpha_n$	$\alpha_h$	$\alpha_b$	$\alpha_n$	$\alpha_h$	$\alpha_b$	$\alpha_n$	$\alpha_h$	$\alpha_b$	$\alpha_n$	$\alpha_h$	#
<b>ZS-CLIP</b>	95.6	95.5	95.5	29.5	34.1	31.6	67.1	76.5	71.5	81.6	68.0	74.2	68.4	68.5	68.2	-
<b>PromptFL</b>	96.6 $\pm$ 0.3	95.6 $\pm$ 0.4	96.1 $\pm$ 0.3	32.2 $\pm$ 1.2	34.6 $\pm$ 1.0	33.4 $\pm$ 1.1	73.2 $\pm$ 0.8	74.9 $\pm$ 0.8	74.0 $\pm$ 0.3	87.4 $\pm$ 0.1	69.1 $\pm$ 1.1	77.2 $\pm$ 0.7	72.4	68.5	70.2	-
<b>FedOTP</b>	97.3 $\pm$ 0.2	95.4 $\pm$ 0.8	96.3 $\pm$ 0.4	32.5 $\pm$ 1.8	34.7 $\pm$ 2.2	33.5 $\pm$ 0.5	72.1 $\pm$ 0.3	74.5 $\pm$ 0.1	73.3 $\pm$ 0.1	86.5 $\pm$ 2.5	70.5 $\pm$ 0.8	77.6 $\pm$ 0.6	72.1	68.8	70.2	<u>3</u>
<b>FedTPG</b>	96.7 $\pm$ 0.3	95.7 $\pm$ 0.6	96.2 $\pm$ 0.4	32.8 $\pm$ 0.4	33.2 $\pm$ 0.4	33.0 $\pm$ 0.2	73.0 $\pm$ 0.1	75.9 $\pm$ 0.3	74.4 $\pm$ 0.3	85.8 $\pm$ 0.4	68.0 $\pm$ 0.3	75.9 $\pm$ 0.3	72.1	67.8	69.9	2
<i>f-CoCoOp</i>	96.5 $\pm$ 0.2	96.0 $\pm$ 0.6	96.2 $\pm$ 0.3	30.4 $\pm$ 2.6	35.8 $\pm$ 1.0	32.9 $\pm$ 2.6	71.7 $\pm$ 0.4	75.3 $\pm$ 0.3	73.5 $\pm$ 0.4	84.0 $\pm$ 1.5	72.1 $\pm$ 1.0	77.6 $\pm$ 0.2	69.7	67.3	68.1	2
<i>f-PLOT</i>	96.5 $\pm$ 0.4	95.7 $\pm$ 0.3	96.1 $\pm$ 0.3	32.7 $\pm$ 0.8	34.7 $\pm$ 0.7	33.7 $\pm$ 0.8	71.7 $\pm$ 0.1	76.2 $\pm$ 0.1	73.9 $\pm$ 0.1	88.4 $\pm$ 4.3	68.6 $\pm$ 2.5	77.2 $\pm$ 1.2	54.4	49.8	51.7	2
<i>f-ProDA</i>	96.7 $\pm$ 0.4	95.1 $\pm$ 1.1	<b>95.9</b> $\pm$ 0.6	31.7 $\pm$ 1.1	35.7 $\pm$ 1.1	33.5 $\pm$ 0.8	72.6 $\pm$ 0.6	74.7 $\pm$ 0.5	73.6 $\pm$ 0.1	86.0 $\pm$ 2.5	70.3 $\pm$ 1.5	77.3 $\pm$ 0.7	71.7	69.0	70.1	2
<i>f-ProGrad</i>	96.8 $\pm$ 0.3	96.2 $\pm$ 0.4	<u>96.5</u> $\pm$ 0.4	32.4 $\pm$ 0.5	34.4 $\pm$ 1.2	33.4 $\pm$ 0.8	72.0 $\pm$ 0.5	76.5 $\pm$ 0.6	74.2 $\pm$ 0.1	86.6 $\pm$ 2.0	70.3 $\pm$ 0.8	77.6 $\pm$ 0.7	72.0	69.4	70.4	<b>4</b>
<i>f-SRC</i>	96.7 $\pm$ 0.1	95.8 $\pm$ 0.2	96.2 $\pm$ 0.1	32.2 $\pm$ 1.0	35.5 $\pm$ 0.8	<u>33.8</u> $\pm$ 0.7	72.4 $\pm$ 0.3	77.0 $\pm$ 0.2	<u>74.6</u> $\pm$ 0.1	86.4 $\pm$ 0.5	73.4 $\pm$ 0.5	<u>79.4</u> $\pm$ 0.5	71.9	70.4	<u>71.0</u>	<b>4</b>
<i>f-KgCoOp</i>	96.7 $\pm$ 0.6	96.0 $\pm$ 0.2	96.3 $\pm$ 0.3	33.4 $\pm$ 0.5	34.3 $\pm$ 1.0	<u>33.8</u> $\pm$ 0.6	72.9 $\pm$ 1.0	75.9 $\pm$ 0.2	74.3 $\pm$ 0.5	88.0 $\pm$ 2.1	70.6 $\pm$ 0.4	78.3 $\pm$ 0.7	72.8	69.2	70.7	<b>4</b>
<i>f-MaPLe</i>	98.4 $\pm$ 0.3	97.0 $\pm$ 0.4	<b>97.6</b> $\pm$ 0.3	34.6 $\pm$ 0.3	35.9 $\pm$ 0.2	<b>35.2</b> $\pm$ 0.3	74.2 $\pm$ 0.5	77.1 $\pm$ 0.3	<b>75.6</b> $\pm$ 0.4	88.9 $\pm$ 0.9	72.7 $\pm$ 0.8	<b>80.0</b> $\pm$ 0.8	74.0	70.7	<b>72.1</b>	<b>4</b>

### D.3 Social Impact

On the positive side, our benchmark results shed lights on the suitable application scenarios of each FPL algorithm. This allows more efficient model adaptation without centralized data collection, reducing risks of sensitive data exposure. This also promotes AI applications relying on multi-modal models in privacy-sensitive scenarios, *e.g.*, assistive technology for disabilities, federated medical imaging analysis. On the negative side, if personalized datasets in the FL network are homogenous or skewed, the personalized model via FPL algorithms may perpetuate or amplify biases (*e.g.*, cultural stereotypes) of pretrained models.