A Experimental Details and Additional Results

A.1 Experimental Details

A.1.1 Non-IID Settings.

(a) Feature shift non-IID (b) Label shift non-IID (c) Feature & label shift non-IID

Figure 4: Illustration of three non-IID settings on Digit-5 dataset. Each dot represents a set of samples within a certain class allocated to a client. The feature shift non-IID is indicated by dot colors while the label shift non-IID ($\alpha = 1$) is indicated by dot sizes.

To mimic non-IID scenarios in a more general way, we investigate three different non-IID settings as follows and visualize them in Figure 4.

- *Feature shift* non-IID: The datasets owned by clients have the same label distribution but different feature distributions. The number of classes and the number of samples per class are the same across clients.
- *Label shift* non-IID: The datasets owned by clients have the same feature distribution but different label distributions. Similar to existing works [69, 78], we use Dirichlet distribution with parameter α to allocate examples for this kind of non-IID setting.
- *Feature & Label shift* non-IID: The datasets owned by clients are different in both label distribution and feature distribution, which is more common but challenging in real-world scenarios.

(a) Digit-5 (b) Office-10 (c) DomainNet

Figure 5: Examples of raw instances from three datasets: Digit-5 (*left*), Office-10 (*middle*), and DomainNet (*right*). We present five classes for each dataset to show the feature shift across their sub-datasets.

A.1.2 Visualizatin of Raw Samples.

Some examples of raw instances can be found in Figure 5.

A.1.3 Model Architecture.

For the single backbone cases, we use ResNet18 pre-trained on Quickdraw as the backbone. For the multiple backbone cases, we use three pre-trained ResNet18 as the backbones. They are pre-trained on Quick Draw [79], Aircraft [80], and CU-Birds [19] public dataset, respectively. The model architecture following the backbone module is shown in Table 7.

Table 7: The model architecture with learnable parameters for each client. FC refers to fully connected layer and BN refers to the BatchNormalization layer. K refers to the number of available pre-trained backbones, which is 3 in our experiments.

Layer	Details
1	FC(512$\times K$, 256), ReLU, BN(256)
2	FC(256, 10)

A.1.4 Training Details.

We provide the detailed settings for the experiments conducted in Section 5.2.

For data splitting, we use a portion of data as training samples ($\sim 10\%$) and the rest as test samples. We first take out a 20% subset of the training set for validation and return the validation set back to the training set and retrain the model after selecting the optimal hyperparameters.

Table 8, 9, 10 show the data partitioning details in feature shift non-IID, label shift non-IID, and feature & label shift non-IID, respectively. We run each algorithm till the convergence of its loss.

Table 8: Detailed statistics for three benchmark datasets in feature shift non-IID, label IID setting (Table 2).

Datasets	MNIST	SVHN	USPS	SynthDigits	MNIST-M
# of clients	1	1	1	1	1
# of classes per client	10	10	10	10	10
# of samples per class	10	10	10	10	10

Table 9: Detailed statistics for three benchmark datasets in label shift non-IID, feature IID setting (Table 11).

Datasets	MNIST-M	Caltech	Real
# of clients	5	5	5
# of samples per client	[152,92,112,72,70]	[76,79,111,70,112]	[153,95,111,55,84]

Table 10: Detailed statistics for three benchmark datasets in feature & label shift non-IID setting (Table 11).

Datasets	Digit-5	Office-10	DomainNet
# of clients	5	4	6
# of samples per client	[100,75,112,120,65]	[89,108,58,120]	[137,230,270,204,175,152]

For hyperparameter tuning, we use grid search to find the optimal hyperparameters including learning rate, weight decay, and the output dimension of each backbone. Concretely, the grid search is carried out in the following search space:

- learning rate: $\{0.1, 0.01, 0.001, 0.0001\}$
- weight decay: {1e-3, 5e-4, 1e-4, 5e-5, 1e-5}
- output dimension of each backbone: $\{128, 256, 512, 1024\}$

A.2 Additional Experiments

A.2.1 Performance under Three Non-IID Settings.

Table 11 shows the performance of FedPCL and baseline methods under label shift non-IID scenarios.

A.2.2 Fairness across Clients.

Following the metrics in [81], we verify the advantage of FedPCL in fairness and report the results over 40 and 80 clients in Table 12 and Table 13, respectively. We list the average, the worst 10%,

Table 11: Test accuracy under the (1) label shift non-IID setting, (2) feature & label shift non-IID setting. For the former (feature IID, label non-IID), we use MNIST-M, Caltech, and Real as the datasets of all clients, respectively. For the latter (feature non-IID, label non-IID), we use Digit-5, Office-10, and DomainNet as the datasets, respectively.

Feature	Method	Single backbone			Multi-backbone		
		Digit-5	Office-10	Domainnet	Digit-5	Office-10	DomainNet
IID	FedAvg	30.42(5.34)	28.84(1.01)	25.78(1.48)	31.97(3.48)	23.60(1.72)	28.09(2.91)
	pFedMe	28.14(1.26)	22.53(5.28)	29.43(1.30)	32.82(1.74)	25.73(3.26)	32.65(0.72)
	Per-FedAvg	27.22(7.11)	36.74(3.96)	31.37(3.35)	30.95(2.94)	25.10(0.55)	34.64(0.54)
	FedRep	34.14(7.37)	36.35(0.53)	**41.95**(3.35)	39.27(1.35)	37.95(0.91)	48.82(0.55)
	FedProto	38.02(3.89)	39.04(0.13)	34.41(0.74)	42.17(3.23)	40.78(0.93)	44.48(0.58)
	Solo	36.40(3.71)	37.82(1.79)	39.06(3.27)	41.33(2.22)	39.59(2.49)	46.70(0.75)
	Ours	**40.88**(2.09)	**40.76**(0.21)	39.63(3.31)	**45.35**(1.58)	**42.13**(0.77)	**52.92**(3.47)
Non-IID	FedAvg	31.03(4.29)	25.67(1.79)	16.40(1.25)	32.98(3.44)	33.84(4.59)	19.25(2.03)
	pFedMe	25.13(8.31)	22.65(1.10)	16.56(2.37)	28.10(8.80)	30.00(1.41)	18.46(2.04)
	Per-FedAvg	28.94(0.60)	25.87(0.19)	18.68(1.47)	31.95(1.81)	26.04(1.46)	20.96(1.83)
	FedRep	34.16(3.40)	32.50(2.86)	19.64(1.53)	39.28(2.44)	37.24(1.54)	30.28(1.01)
	FedProto	41.49(2.75)	37.47(1.27)	19.37(0.14)	43.94(3.02)	34.54(2.65)	29.45(0.39)
	Solo	35.56(3.16)	35.54(1.05)	17.84(0.87)	38.35(2.55)	36.38(0.54)	27.15(0.52)
	Ours	**42.87**(1.47)	**37.64**(1.99)	**24.90**(1.61)	**45.22**(3.65)	**41.40**(1.19)	**35.23**(0.29)

the worst 20%, the worst 40%, the best 10% of test accuracy over all clients across three runs in Digit-5 dataset. We also report the variance of the accuracy distribution across clients (the smaller, the fairer) [81]. The results demonstrate that in such a multi-backbone scenario, it is easier to obtain a fair solution using prototype-based communication rather than model parameter/gradient-based communication, because the information conveyed by prototypes is more local data distribution-independent while the information conveyed by model parameters tends to be affected by local data distribution. Detailed experimental results on fairness can be found in Table 12 and 13.

Table 12: The average, the worst, the best, and the variance of the test accuracy of 40 clients on Digit-5.

Method	Average	Worst 10%	Worst 20%	Worst 40%	Best 10%	Variance
FedAvg	32.20(2.17)	16.73(1.73)	20.31(2.21)	25.01(0.71)	54.60(3.33)	11.16(0.35)
Ours	48.39(0.25)	35.35(2.63)	37.58(3.07)	40.82(3.16)	62.72(1.33)	8.45(0.06)

Table 13: The average, the worst, the best, and the variance of the test accuracy of 80 clients on Digit-5.

Method	Average	Worst 10%	Worst 20%	Worst 40%	Best 10%	Variance
FedAvg	33.44(3.34)	12.34(2.67)	15.53(2.90)	19.53(2.40)	56.87(2.74)	13.20(1.35)
Ours	49.46(0.34)	30.39(4.57)	34.37(3.40)	39.52(2.66)	66.46(2.38)	10.60(0.25)

A.2.3 Effect of the Number of Backbones.

The number of pre-trained backbones can be adjusted for a specific task correspondingly. To study its effect, we compare the performance and parameter size when different numbers of fixed backbones are used. The results in Table 14 indicate that more pre-trained backbones can lead to better performance but consume more computing resources and memory space.

A.2.4 Privacy Protection.

We incorporate FedPCL with privacy-preserving techniques to observe its variation in performance. Concretely, we add random noise of various distributions into the communicated prototypes and the original images, respectively. Table 15 shows that the performance of FedPCL remains high after injecting noise to the prototypes. Figure 6 visualizes an original image of a bike from Office-10 dataset and what it looks like after the noise injection operation. It is hard to tell the objects in the right column of Figure 6(c), but the accuracy only drops about 2%, compared to vanilla FedPCL.

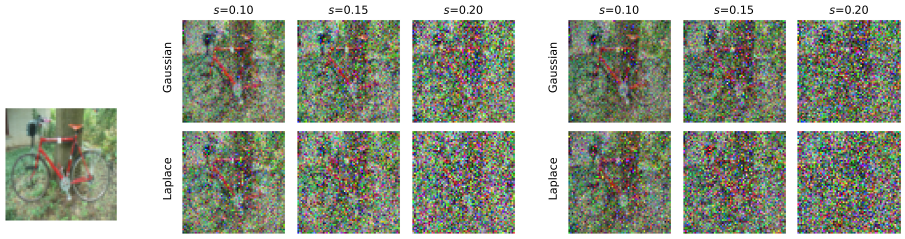
Table 14: Effect of the number of pre-trained backbones. Experiments are conducted on Digit-5 dataset under the feature & label shift non-IID setting where the Dirichlet parameter α is 1, and the number of clients is 5.

# of Backbones	# of Training Params.	# of Fixed Params.	Acc
1	133,632	11M	42.87(1.47)
2	264,704	22M	44.49(1.75)
3	395,776	33M	45.22(3.65)

In conclusion, FedPCL has the potential to combine with privacy-preserving techniques without an obvious decrease in performance.

Table 15: The performance of FedPCL on Office-10 dataset after incorporating privacy-preserving techniques. Here, we consider using multiple backbones under the label shift non-IID setting with $\alpha = 1$ and $m = 5$. s represents the scale parameter for the noise distribution generation and $p \in (0, 1)$ represents the perturbation coefficient of the noise.

Methods	Add Noise to	Noise Type	Acc(Std)
FedRep	/	/	37.95(0.91)
FedPCL	/	/	42.13(0.77)
FedPCL	Prototype	*Laplace* ($s = 0.05, p = 0.1$)	39.93(3.48)
		Gaussian ($s = 0.05, p = 0.1$)	40.04(2.09)
		Laplace ($s = 0.05, p = 0.2$)	40.24(3.01)
		Gaussian ($s = 0.05, p = 0.2$)	41.83(3.57)
	Image	*Laplace* ($s = 0.2, p = 0.1$)	38.29(2.87)
		Gaussian ($s = 0.2, p = 0.1$)	41.10(0.57)
		Laplace ($s = 0.2, p = 0.2$)	36.93(2.33)
		Gaussian ($s = 0.2, p = 0.2$)	40.14(0.78)



(a) Original (b) $p = 0.1$ (c) $p = 0.2$

Figure 6: Visualization of (a) an original image of a bike from Office-10 dataset; (b-c) applying noise injection privacy-preserving techniques to the original image. Specifically, given an original image x and a perturbation coefficient $p \in (0, 1)$, $\widetilde{x} = (1 - p)x + e$ is the image used for training. Here, we provide the results of two kinds of random noise, Gaussian (upper row) and Laplace noise (lower row). The value of p in Figure 6(b) and Figure 6(c) is 0.1 and 0.2, respectively.

B Discussion

B.1 Applying FedPCL to a Wider Range of Scenarios

Since there are many off-the-shelf foundation models available nowadays and there is a trend to use them in CV and NLP communities, it is a promising research direction to integrate those pre-trained foundation models into an FL framework. Compared with training from scratch, using pre-trained models can save more computation and communication resources. Also, it is an interesting direction

to apply FedPCL to some noisy-label federated learning cases to prevent model degradation caused by incorrect labels when training from scratch [82, 83, 84].

Our proposed framework is limited to the cases where pre-trained models are available. This is mainly due to the fact that most types of data, e.g., image, text, graph, have corresponding pre-trained models nowadays. For the cases without pre-trained models, using multiple fixed encoders can be an alternative solution, which is another interesting problem and can be explored in the future.

We do think incorporating pre-trained models into existing learning frameworks is a promising trend in deep learning, especially when models are becoming larger and larger in scale and hard to train from scratch. So far, the idea to utilize pre-trained foundation models has been proposed for CV and NLP tasks and achieved certain improvement [52, 85, 86].

B.2 Comparison to Related Works

This paper has provided novel contributions in terms of (1) integrating pre-trained models into federated learning, (2) proposing a novel algorithm FedPCL which allows clients to share knowledge via prototype-based local contrastive learning.

Although prototypical learning and contrastive learning exist in prior work, they are still based on the learnable parameter aggregation scheme [43, 44, 51] or just use prototypes/contrastive learning to regularize the original local training [24]. Instead of synchronizing learnable parameters, our method allows each client to keep their own local parameters while extracting shared knowledge only by contrastive learning. Prototype is used as the information carrier to achieve that.

Some state-of-the-art PFL methods do not work well under our proposed lightweight framework. Their performance is mainly due to the following two reasons.

- Most of these baselines are designed based on the training-from-scratch framework where there are a large number of learnable parameters. The proposed new setting where most parameters are fixed is not friendly to some PFL methods.
- When there are feature/label shift non-IID across clients, the local performance might be further deteriorated after parameter aggregation. The deterioration can be more significant when only a small number of parameters are locally trained. Since most PFL methods are still based on the parameter aggregation scheme, their performances are inevitably affected.

Performance might be enhanced by decreasing the number of shared parameters, optimizing local training schemes, and introducing fine-tuning procedures during local training, etc.

There is also a branch of FL studying fine-tuning techniques for FL [87, 88]. These works mainly focus on how to adjust the local/global model to improve its representation ability on biased/generic data distribution, while our work utilizes fixed pre-trained foundation models and focuses on improving the fusing ability.

C Proof of Generalization Bound

Theorem C.1. (Generalization Bound of Alg. 1). *Consider an FL system with m clients. Let $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_m$ be the true data distribution and $\widehat{\mathcal{D}}_1, \widehat{\mathcal{D}}_2, \cdots, \widehat{\mathcal{D}}_m$ be the empirical data distribution. Denote the projector head h as the hypothesis from $\mathcal{H}$ and d be the VC-dimension of $\mathcal{H}$. The total number of samples over all clients is N. Then, with probability at least $1-\delta$:*

$$\begin{aligned}&\max_{(\theta_1,\theta_2,\ldots,\theta_m)} \left| \sum_{i=1}^{m} \frac{|D_i|}{N} L_{\mathcal{D}_i}\left(\theta_i; \mathbb{C}, \{\boldsymbol{C}_p\}_{p=1}^m\right) - \sum_{i=1}^{m} \frac{|D_i|}{N} L_{\widehat{\mathcal{D}}_i}\left(\theta_i; \mathbb{C}, \{\boldsymbol{C}_p\}_{p=1}^m\right) \right| \\ &\leq \sqrt{\frac{N}{2}\log\frac{(m+1)|\mathbb{C}|}{\delta}} + \sqrt{\frac{d}{N}\log\frac{eN}{d}}.\end{aligned} \tag{12}$$

Proof. We start from the McDiarmid's inequality that

$$\mathbb{P}\left[g\left(X_1,\ldots,X_n\right) - \mathbb{E}\left[g\left(X_1,\ldots,X_n\right)\right] \geq \epsilon\right] \leq \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^{n} c_i^2}\right) \tag{13}$$

when

$$\sup_{x_1,\ldots,x_n} |g(x_1, x_2, \ldots, x_n) - g(x_1, x_2, \ldots, x_n)| \leq c_i. \tag{14}$$

Eq. 13 equals to

$$\mathbb{P}\left[g(\cdot) - \mathbb{E}\left[g(\cdot)\right] \leq \epsilon\right] \geq 1 - \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^n c_i^2}\right) \tag{15}$$

which means that with probability at least $1 - \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^n c_i^2}\right)$,

$$g(\cdot) - \mathbb{E}\left[g(\cdot)\right] \leq \epsilon. \tag{16}$$

Let $\delta = \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^n c_i^2}\right)$, the above can be rewritten as with probability at least $1 - \delta$,

$$g(\cdot) - \mathbb{E}\left[g(\cdot)\right] \leq \sqrt{\frac{\sum_{i=1}^n c_i^2}{2} \log \frac{1}{\delta}}. \tag{17}$$

For prototype, by substituting $g(\cdot)$ with

$$\max_{(\theta_1,\theta_2,\ldots,\theta_m)} \left(\sum_{i=1}^m \frac{|D_i|}{N} L_{\mathcal{D}_i}(\theta_i;\cdot) - \sum_{i=1}^m \frac{|D_i|}{N} L_{\widehat{\mathcal{D}}_i}(\theta_i;\cdot)\right), \tag{18}$$

we can obtain that with probability at least $1 - \delta$, the following holds for a specific prototype,

$$\begin{aligned} &\max_{(\theta_1,\theta_2,\ldots,\theta_m)} \left(\sum_{i=1}^m \frac{|D_i|}{N} L_{\mathcal{D}_i}(\theta_i;\cdot) - \sum_{i=1}^m \frac{|D_i|}{N} L_{\widehat{\mathcal{D}}_i}(\theta_i;\cdot)\right) \\ -&\mathbb{E}\left[\max_{(\theta_1,\theta_2,\ldots,\theta_m)} \left(\sum_{i=1}^m \frac{|D_i|}{N} L_{\mathcal{D}_i}(\theta_i;\cdot) - \sum_{i=1}^m \frac{|D_i|}{N} L_{\widehat{\mathcal{D}}_i}(\theta_i;\cdot)\right)\right] \leq \sqrt{\frac{N}{2}\log\frac{1}{\delta}}, \end{aligned} \tag{19}$$

Considering there are $(m+1)|\mathbb{C}|$ prototypes in total, by using Boole's inequality, with probability at least $1 - \delta$, the following holds,

$$\begin{aligned} &\max_{(\theta_1,\theta_2,\ldots,\theta_m)} \left(\sum_{i=1}^m \frac{|D_i|}{N} L_{\mathcal{D}_i}\left(\theta_i;\mathbb{C},\{\boldsymbol{C}_p\}_{p=1}^m\right) - \sum_{i=1}^m \frac{|D_i|}{N} L_{\widehat{\mathcal{D}}_i}\left(\theta_i;\mathbb{C},\{\boldsymbol{C}_p\}_{p=1}^m\right)\right) \\ \leq&\mathbb{E}\left[\max_{(\theta_1,\theta_2,\ldots,\theta_m)} \left(\sum_{i=1}^m \frac{|D_i|}{N} L_{\mathcal{D}_i}\left(\theta_i;\mathbb{C},\{\boldsymbol{C}_p\}_{p=1}^m\right) - \sum_{i=1}^m \frac{|D_i|}{N} L_{\widehat{\mathcal{D}}_i}\left(\theta_i;\mathbb{C},\{\boldsymbol{C}_p\}_{p=1}^m\right)\right)\right] \\ &+\sqrt{\frac{N}{2}\log\frac{(m+1)|\mathbb{C}|}{\delta}}, \end{aligned} \tag{20}$$

where N is the total number of samples over all clients.

$$\begin{aligned} &\mathbb{E}\left[\max_{(\theta_1,\theta_2,\ldots,\theta_m)} \left(\sum_{i=1}^m \frac{|D_i|}{N} L_{\mathcal{D}_i}\left(\theta_i;\mathbb{C},\{\boldsymbol{C}_p\}_{p=1}^m\right) - \sum_{i=1}^m \frac{|D_i|}{N} L_{\widehat{\mathcal{D}}_i}\left(\theta_i;\mathbb{C},\{\boldsymbol{C}_p\}_{p=1}^m\right)\right)\right] \\ \leq&\mathbb{E}\left[\sum_{i=1}^m \frac{|D_i|}{N} \max_{\theta_i} \left(L_{\mathcal{D}_i}\left(\theta_i;\mathbb{C},\{\boldsymbol{c}_p\}_{p=1}^m\right) - L_{\widehat{\mathcal{D}}_i}\left(\theta_i;\mathbb{C},\{\boldsymbol{c}_p\}_{p=1}^m\right)\right)\right] \\ \overset{(a)}{\leq}&\sum_{i=1}^m \frac{|D_i|}{N} \Re_i(\mathcal{H}) \\ \leq&\sum_{i=1}^m \frac{|D_i|}{N} \sqrt{\frac{d}{|D_i|}\log\frac{e|D_i|}{d}} \\ \leq&\sum_{i=1}^m \frac{|D_i|}{N} \sqrt{\frac{d}{|D_i|}\log\frac{eN}{d}} \\ \overset{(b)}{\leq}&\sqrt{\frac{d}{N}\log\frac{eN}{d}} \end{aligned} \tag{21}$$

where $\mathcal{H}$ is the hypothesis set of projector head h, d is the VC-dimension of $\mathcal{H}$, N is the total number of samples over all clients. (a) follows from the definition of Rademacher complexity

$$\widehat{\Re}_S(\mathcal{G}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i g\left(z_i\right)\right] \tag{22}$$

where m is the number of samples, σ_i refers to independent uniform random variable taking value in $\{-1, +1\}$, and (b) follows from Jensen's inequality.

So,

$$\begin{aligned}&\max_{(\theta_1,\theta_2,\ldots,\theta_m)} \left|\sum_{i=1}^{m} \frac{|D_i|}{N} L_{\mathcal{D}_i}\left(\theta_i; \mathbb{C}, \{\boldsymbol{C}_p\}_{p=1}^{m}\right) - \sum_{i=1}^{m} \frac{|D_i|}{N} L_{\widehat{\mathcal{D}}_i}\left(\theta_i; \mathbb{C}, \{\boldsymbol{C}_p\}_{p=1}^{m}\right)\right| \\ &\leq \sqrt{\frac{N}{2}\log\frac{(m+1)|\mathbb{C}|}{\delta}} + \sqrt{\frac{d}{N}\log\frac{eN}{d}}.\end{aligned} \tag{23}$$

□