Prompt Estimation from Prototypes for Federated Prompt Tuning of Vision Transformers

Anonymous authors
Paper under double-blind review

Abstract

Visual Prompt Tuning (VPT) of pre-trained Vision Transformers (ViTs) has proven highly effective as a parameter-efficient fine-tuning technique for adapting large models to downstream tasks with limited data. Its parameter efficiency makes it particularly suitable for Federated Learning (FL), where both communication and computation budgets are often constrained. However, global prompt tuning struggles to generalize across heterogeneous clients, while personalized tuning overfits to local data and lacks generalization. We propose PEP-FedPT (Prompt Estimation from Prototypes for Federated Prompt Tuning), a unified framework designed to achieve both generalization and personalization in federated prompt tuning of ViTs. Within this framework, we introduce the novel Class-Contextualized Mixed Prompt (CCMP) — based on class-specific prompts maintained alongside a globally shared prompt. For each input, CCMP adaptively combines class-specific prompts using weights derived from global class prototypes and client class priors. This approach enables per-sample prompt personalization without storing client-dependent trainable parameters. The prompts are collaboratively optimized via traditional federated averaging technique on the same. Comprehensive evaluations on CIFAR-100, TinyImageNet, DomainNet, and iNaturalist datasets demonstrate that PEP-FedPT consistently surpasses the state-of-the-art baselines under diverse data heterogeneity scenarios, establishing a strong foundation for efficient and generalizable federated prompt tuning of Vision Transformers.

1 Introduction

Federated learning (FL) (McMahan et al., 2017) is a collaborative machine learning approach in which a central server coordinates multiple clients to jointly train a global model, while preserving the privacy of the client's data by keeping the local data decentralized. A major challenge in FL is data heterogeneity: datasets on each client can differ significantly in distribution, resulting in non-identically distributed (non-iid) data across the network. These discrepancies often cause client models to converge to different local minima, a phenomenon known as "client drift" (Karimireddy et al., 2020), which in turn degrades the generalization performance of the global model. To address this issue, personalized FL methods (Chen & Chao; Ma et al., 2022; Shamsian et al., 2021) have been proposed, aiming to better accommodate diverse local data distributions. However, a key drawback of personalized methods is their reduced generalization performance on new or unseen clients (Deng et al., 2024).

Inspired by their success in centralized learning, large foundation models (FMs) are increasingly adopted in FL to mitigate data heterogeneity (Bommasani et al., 2021; Dosovitskiy et al., 2021; Radford et al., 2021). FMs demonstrate enhanced robustness in non-iid data settings (Qu et al., 2022). Yet, their substantial computational demands for tuning on resource-constrained edge devices and high communication costs present major hurdles. Parameter-efficient tuning methods, such as Visual Prompt Tuning (VPT) (Jia et al., 2022), offer a promising solution for efficient FM adaptation within FL, significantly reducing communication overhead while harnessing the power of large models.

Prompt tuning has recently gained attention in Federated Learning (FL), with methods like FedPR (Feng et al., 2023) and pFedPG (Yang et al., 2023) exploring its potential. However, each comes with notable limitations. FedPR employs global prompts and is primarily suited for cross-silo FL. This design struggles in highly heterogeneous client settings, where shared global prompts fail to adapt to diverse local data distributions (Li et al., 2020; Deng et al., 2024). pFedPG, on the other hand, generates client-specific prompts at the server and sends them to clients for local fine-tuning. Although personalized, this approach implicitly assumes full client participation in every round—an impractical assumption in many real-world FL systems. Moreover, such personalized strategies risk fitting to local data (Wu et al.; Deng et al., 2024), limiting their generalization to unseen or non-participating clients (as seen in Table 3). To address these issues, SGPT (Deng et al., 2024) employs shared and group-specific prompts to enhance generalization. However, its two-stage training process and non-differentiable mechanism add optimization complexity and computational overhead. Furthermore, when data heterogeneity is high, it still struggles to generalize across diverse client distributions (Tables 1 and 2). These limitations highlight a key trade-off in FL prompt tuning: global prompts generalize well but lack expressiveness, while personalized prompts offer local adaptability but suffer from poor generalization and scalability. This raises the central question:

Can we achieve effective personalization while relying solely on globally shared prompts?

We answer this by proposing a novel prompt-tuning strategy tailored for fine-tuning of vision transformer in FL. Our method introduces class-specific prompts that are jointly optimized with shared prompts to address data heterogeneity across clients. To induce personalization without local prompt storage, we propose Prompt Estimation from Prototypes for Federated Prompt Tuning (PEP-FedPT). It generates a Class-Contextualized Mixed Prompt (CCMP) by combining global class-specific prompts. The combination weights are determined by per-class membership scores, computed using global `cls`-token prototypes and the client's local class priors. The global `cls` prototype aggregates class centroids across clients, where each centroid is computed by averaging the `cls` token representations of data points within a given class. For a given input, we estimate class membership scores—refined by each client's class priors. These scores act as soft weights to combine class-specific prompts into a single, differentiable mixed prompt (CCMP). This allows each client to dynamically personalize prompts using only shared global information, without the need for local prompt storage or specialized server-side generation. Our approach integrates seamlessly into standard FL pipelines and delivers strong empirical performance across heterogeneous datasets, as demonstrated in Tables 1 and 2. Despite relying solely on global prompts, it effectively utilizes clients' data distribution, achieving scalable generalization.

Our key contributions are as follows:

1. We introduce a unified framework PEP-FedPT that jointly optimizes class-specific and shared prompts to address data heterogeneity in federated learning of ViTs. We exploit the clients' distribution to achieve personalization by using only global prompts. Thus our proposed strategy aims to strike an effective balance between generalizability and personalization.

2. We design a novel prompt-mixing strategy that generates Class Contexualized Mixed Prompts (CCMP) as a function of class-specific prompts shared globally across clients. We empirically demonstrate its superiority over existing methods through extensive experiments on datasets exhibiting feature and label imbalance.

3. We provide theoretical insights into our design by showing that CCMP minimizes a quadratic upper bound and is optimal in the Minimum Mean Squared Error (MMSE) sense.

2 Related Work

2.1 Federated Learning (FL):

FL is a machine learning paradigm that emphasizes data privacy by enabling collaborative model training under the coordination of a central server, without requiring direct data sharing. FedAvg (McMahan et al., 2017) is the most commonly used technique for aggregating local models. This has led to its broad adoption across various domains, such as Internet of Things and mobile devices (Mills et al., 2019; Nguyen et al., 2021; Hard et al., 2018; Ramaswamy et al., 2019), healthcare (Rieke et al., 2020; Xu et al., 2021; Nguyen et al., 2021; Brisimi et al., 2018; Feng et al., 2023), person re-identification (Zhuang et al., 2020), and face recognition (Liu et al., 2022a). Under data-heterogeneity, training the models using FedAvg leads to client-drift . Addressing this, regularization techniques (Acar et al.; Li et al., 2020; Gao et al., 2022) and variance reduction methods (Karimireddy et al., 2020), and several studies on improving the generalization performance in FL by inducing flatness during the local training (Sun et al., 2023; Caldarola et al., 2022) have come to light. In case of extreme data heterogeneity across the clients, training a single model for all the clients will be difficult. As an alternative, personalized FL approaches have been proposed (Tan et al., 2023; Chen & Chao; Ma et al., 2022; Shamsian et al., 2021). These frameworks share some model parameters with the server while keeping others client-specific, enabling adaptation to local data distributions.

2.2 Prompt Tuning and Federated Learning:

As the fine-tuning of the foundational models became ubiquitous for downstream tasks in centralized learning (Bommasani et al., 2021; Dosovitskiy et al., 2021; Radford et al., 2021), prompt tuning techniques were originally proposed in the NLP community (Li & Liang, 2021; Liu et al., 2022b). Recently, Visual Prompt Tuning (VPT) was proposed for prompt tuning in ViT models, demonstrating its efficiency. ViT based FL (Qu et al., 2022) shows robustness to heterogeneity. However, due to heavy communication costs, prompt-tuning on pre-trained VIT models gained attention. In the recent literature, prompt tuning for FL has been proposed in methods like SGPT (Deng et al., 2024), FedPR (Feng et al., 2023), pFedPG (Yang et al., 2023) and also in the context of Vision and Language models FedOTP (Li et al., 2024). SGPT relies on group-specific prompts and requires alternate training due to a non-differentiable selection mechanism. pFedPG depends on full client participation, while FedOTP assumes access to both text and image encoders. We address these limitations by introducing class-specific prompts and a novel per-sample prompt-mixing strategy based on class priors and `cls`-token prototypes.

3 Preliminary

In this paper, we use boldface letters to denote matrices and vectors. The operator "$\cdot$" refers to element-wise multiplication and $*$ refers to the standard matrix multiplication. We define TL_i as the i^{th} transformer layer, and $\mathbb{E}$ denotes the expectation operator. The term δ_k^c represents the probability of observing samples from class c at client k. Additionally, $sim(\mathbf{p}, \mathbf{q})$ denotes the cosine similarity, while $\mathbb{I}$ represents the indicator function. $[M]$ denotes the set $\{1, 2, ., .M\}$[1].

3.1 Visual Prompt Tuning (VPT)

VPT is a parameter-efficient fine-tuning method for the pre-trained ViT (Jia et al., 2022). It is an efficient alternative to fine-tuning the full model. Jia et al. (2022) propose VPT, where prompts are inserted at the input of the ViT. with d dimensional trainable prompt $\mathbf{P}_0 \in \mathbb{R}^{d\times 1}$ as follows:

$$\mathbf{cls}_i, \mathbf{P}_i, \mathbf{E}_i = TL_i([\mathbf{cls}_{i-1}, \mathbf{P}_{i-1}, \mathbf{E}_{i-1}]) \tag{1}$$

$$\mathbf{y} = \mathbf{H} * \mathbf{cls}_M \tag{2}$$

$\mathbf{E_i} \in \mathbb{R}^{d\times n_I}$ denotes the image tokens at layer i, n_I denotes the number of image tokens, M denotes the number of layers in the transformer. The final layer's `cls` token i.e, $\mathbf{cls}_M$ is used for classification. In the above model, the classification head $\mathbf{H}$ and $\mathbf{P}_0$ are trainable.

3.2 Federated Learning (FL)

In FL, the server orchestrates the training with n clients with the goal of minimizing the following training objective:

$$\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) := \frac{1}{n}\sum_{k=1}^{n} f_k(\boldsymbol{\theta}) \tag{3}$$

[1] The detailed notations and definitions are in Sec. A.1 of Appendix

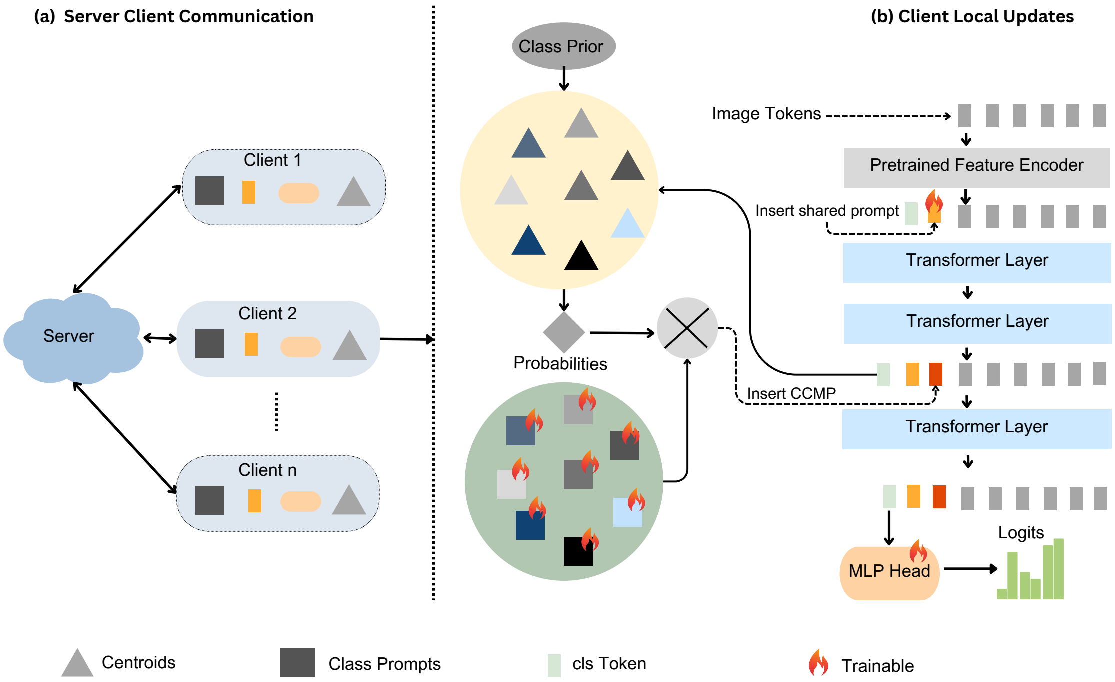


Figure 1: The left panel (a) illustrates server-client communication during federated training. In each communication round, clients (right panel (b)) insert shared prompts at the input of the transformer and class-contextualized prompts— derived by mixing class prompts using probabilities computed from local class priors, cls-tokens and centroids—at intermediate layer(s).

f_k denotes the k^{th} client local objective function. $\boldsymbol{\theta}$ denotes the model parameters shared across the clients. In general, it can be written as $f_k(\boldsymbol{\theta}) = \underset{(\mathbf{x},y)\sim\mathcal{D}_k}{\mathbb{E}} l_k(\boldsymbol{\theta};(\mathbf{x},y))$. $\mathcal{D}_k$ denotes the data distribution of the client k and $l_k(\boldsymbol{\theta};(\mathbf{x},y))$ denotes the task-specific loss function. For a classification task, $\mathbf{x}$ denotes the input and y is the ground truth. In FL training, at each round t, the server broadcasts the global model $\boldsymbol{\theta}^t$ to a randomly selected subset of clients S_t. Each client $k \in S_t$ performs several steps of local training starting from $\boldsymbol{\theta}^t$, and then sends its updated model $\boldsymbol{\theta}_k^t$ back to the server. The server aggregates these updates using federated averaging:

$$\boldsymbol{\theta}^{t+1} = \frac{1}{|S_t|} \sum_{k\in S_t} \boldsymbol{\theta}_k^t.$$

The updated model $\boldsymbol{\theta}^{t+1}$ is then broadcast to clients in the next round. This procedure describes the basic FedAvg algorithm McMahan et al. (2017).

4 Proposed Method: Prompt Estimation from Prototypes- Federated Prompt Tuning (PEP-FedPT)

We consider a federated learning setup with n clients coordinated by a central server, where each client's data is drawn from a distinct distribution $\mathcal{D}_k$. Following VPT (Jia et al., 2022), we assume each client uses a pre-trained ViT-B/16 as its local model architecture. We introduce Shared Prompts and Class-Contextualized Mixed Prompts (CCMP). Also, by utilizing the information in the local class priors and the global class prototypes, we softly combine the class-specific prompts, leading to per-client customization while sharing the global class-specific prompts.

4.1 Prompt Design

We propose two kinds of prompts: Shared Prompts $\mathbf{P}_S$, and Class Contextualized Mixed Prompts (CCMP) denoted by $\mathbf{m}$ which is a function of Class Specific Prompts $\mathbf{P}_C$. The Shared Prompts $\mathbf{P}_S$ and the Class Prompts $\mathbf{P}_C$ are common across the clients and are shared with the server. In our methodology, the input and output of ViT for a layer i' can be compactly written as below

$$\mathbf{cls}_{i'}, \mathbf{m}_{i'}, \mathbf{P}_{S_{i'}}, \mathbf{E}_{i'} = TL_{i'}(\mathbf{cls}_{i'-1}, \mathbf{m}_{i'-1}, \mathbf{P}_{S_{i'-1}}, \mathbf{E}_{i'-1}) \tag{4}$$

In the Eq. 4, $\mathbf{m}_{i'-1}, \mathbf{P}_{S_{i'-1}}$ denotes the tokens corresponding CCMP and the shared prompts respectively at the input of layer i'. We now describe each of these prompts in detail.

Shared Prompts ($\mathbf{P}_S$): Inspired by Jia et al. (2022) we added shared prompts at the very first layer of the ViT model. These are shared across the clients and are given by $\mathbf{P}_S = \left[\mathbf{p}_{s_1}\mathbf{p}_{s_2}...\mathbf{p}_{s_{|S|}}\right]$, where $|S|$ is the number of shared prompts inserted and are processed as follows:

$$\mathbf{cls}_1, \mathbf{P}_{S_1}, \mathbf{E}_1 = TL_1([\mathbf{cls}_0, \mathbf{P}_S, \mathbf{E}_0]) \tag{5}$$

As discussed in Ostapenko et al. (2022) and Deng et al. (2024), the early layers capture the low-level representations and can be shared across the classes. This allows the model to have better generalization. It is shown that the representations in the initial layer of pre-trained ViT are uniformly distributed on the manifold, indicating that the information is shared across the classes. This is true even when the data distribution is different across the clients as shown Sec A.5.4 of the appendix.

Class Contextualized Mixed Prompts (CCMP) ($\mathbf{m}(k)$): This is a client-specific prompt, and it is obtained by softly combining the class-specific prompts given by $\mathbf{P}_C = \left[\mathbf{p}_{c_1}\mathbf{p}_{c_2}...\mathbf{p}_{c_{|C|}}\right]$, where $|C|$ is equal to the total number of classes and $\mathbf{P}_C \in \mathbb{R}^{d\times|C|}$. These class-specific prompts $\mathbf{P}_C$ are shared across the clients, but the weights used to combine these class-specific prompts are local to each client. These soft weights for a client k at the input of layer j on the i-th training input denoted by $\mathbf{s}_{i,j-1,k} \in [0,1]^{|C|\times 1}$ are designed as the function of input data point $\mathbf{x}_{i,k}$, `cls` token prototypes and class priors. Finally, the CCMP $\mathbf{m}_{j-1}$ is added at the input of layer j, and it's given below

$$\mathbf{m}_{j-1}(k) = \mathbf{P}_C * \mathbf{s}_{i,j-1,k} \tag{6}$$

The overall input and output after adding the CCMP $\mathbf{m}_{j-1}$ at the input of layer j is shown below:

$$\mathbf{cls}_j, \mathbf{m}_j(k), \mathbf{P}_{S_j}, \mathbf{E}_j = TL_j([\mathbf{cls}_{j-1}, \mathbf{m}_{j-1}(k), \mathbf{P}_{S_{j-1}}, \mathbf{E}_{j-1}]). \tag{7}$$

The soft weights are explained in Sec. 4.2. If the ViT has M layers, the final logits are given by:

$$\mathbf{y} = \mathbf{H} * \mathbf{cls}_M \tag{8}$$

$\mathbf{H}$ denotes the classification layer. Finally, we aim to solve the following federated optimization problem involving the shared and class-specific prompts.

$$\min_{\mathbf{P}_S,\mathbf{P}_C,\mathbf{H}} \frac{1}{n}\sum_{k=1}^{n} f_k(\mathbf{w}_{pre}; \mathbf{P}_S, \mathbf{P}_C, \mathbf{H}) \tag{9}$$

Here $\mathbf{w}_{pre}$ denotes the pre-trained ViT parameters and $\mathbf{P}_S, \mathbf{P}_C$ denote the shared, class-specific prompts, respectively and f_k denotes the loss of the client. Here $\mathbf{P}_S, \mathbf{P}_C$ are shared between clients.

4.2 Estimation of Soft Weights for CCMP

We present the design of the soft weights, which are aimed to provide class-specific information to the model. We exploit the information present in the class prototypes of $\mathbf{cls}_{l-1}$ token at a layer l. Our empirical observations show that `cls` tokens of the pre-trained ViT model carry significant information regarding the downstream task. In the Figure 2 we observe the Top-5 zero-shot test accuracy of the CIFAR-100 dataset computed at each layer. The accuracy at layer l is computed by taking the minimum distance between the `cls` token corresponding to the test input and the class prototypes of the `cls` token at the input of layer l.

We now describe the soft weights computed for a layer l i.e, $\mathbf{s}_{l-1}$. Let us denote the `cls` token at the input of layer l corresponding to data point $\mathbf{x}_{i,k}$ for client k in communication round t as $\mathbf{cls}_{l-1,i,k,t}$. The `cls` token's class prototype for the class c at communication round t is denoted by $\boldsymbol{\mu}^c_{l-1,k,t}$ and it is computed as in Eq. 10

$$\boldsymbol{\mu}^c_{l-1,k,t} = \begin{cases} \frac{1}{n_{k,c}} \sum_{i=1}^{N_k} \mathbf{cls}_{l-1,i,k,t} \cdot \mathbb{I}_{y_{i,k}=c}, & n_{k,c} > 0, \\ \mathbf{0}, & n_{k,c} = 0. \end{cases} \tag{10}$$

Here $n_{k,c} = \sum_{i=1}^{N_k} \mathbb{I}_{y_{i,k}=c}$, where N_k denotes the number of data points of client k, and $\mathbb{I}_{y_{i,k}=c}$ denotes indicator function. It takes value 1 if the data point i of client k belongs to class c otherwise it is 0.

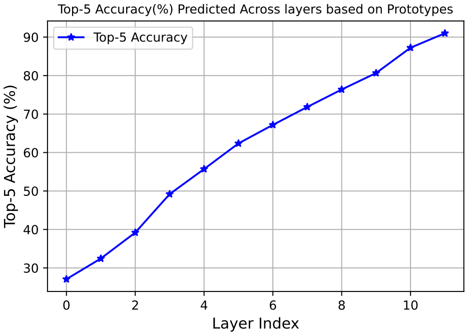


Figure 2: The Top-5 accuracy computed based on the minimum distance between the cls token corresponding to the input and the cls prototypes. This shows that the `cls` representations in the middle layers have coarse information of the task.

After every fixed update period R, the server aggregates the prototypes from the clients to compute the aggregated prototype $\hat{\boldsymbol{\mu}}^c_{l-1,r}$ at the r-th period as Eq. 11. Let the set of communication rounds within this update period be $\Lambda = \{rR, rR+1, \ldots, (r+1)R-1\}$.

$$\hat{\boldsymbol{\mu}}^c_{l-1,r} = \begin{cases} \frac{1}{D_c} \sum_{t\in\Lambda} \sum_{k\in S_t} \mathbb{I}_{\{\boldsymbol{\mu}^c_{l-1,k,t} \neq \mathbf{0}\}} \boldsymbol{\mu}^c_{l-1,k,t}, & D_c > 0, \\ \mathbf{0}, & D_c = 0. \end{cases} \tag{11}$$

here $D_c = \sum_{t\in\Lambda} \sum_{k\in S_t} \mathbb{I}_{\{\boldsymbol{\mu}^c_{l-1,k,t} \neq \mathbf{0}\}}$, $S_t \subseteq [n]$ denotes the subset of clients sampled by the server at a round t. As the training progresses, the server uses the momentum to update the aggregated prototypes $\hat{\boldsymbol{\mu}}^c_{l-1,r}$ to form the global class prototype $\boldsymbol{\mu}^c_{l-1,r}$ as in Eq. 12 which is then communicated to the clients. The parameter ρ denotes the momentum. The updated prototypes $\boldsymbol{\mu}^c_{l-1,r}$ are sent to the clients.

$$\boldsymbol{\mu}^c_{l-1,r} = \rho \cdot \boldsymbol{\mu}^c_{l-1,r-1} + (1-\rho) \cdot \hat{\boldsymbol{\mu}}^c_{l-1,r} \tag{12}$$

If D_c is 0, we set $\rho = 1$.

We now define the un-normalized score function $\hat{s}^c_{i,l-1,k}$ assigned by the cls token $\mathbf{cls}_{l-1,i,k}$, corresponding to input $\mathbf{x}_{i,k}$, to class-specific prompt $\mathbf{p}_c$ at the client k. Here we drop the index of the communication round t and the update period r for better readability.

$$\hat{s}^c_{i,l-1,k} = exp\left(\frac{sim\left(\mathbf{cls}_{l-1,i,k}, \boldsymbol{\mu}^c_{l-1}\right)}{\tau}\right) \delta^c_k \tag{13}$$

We define $sim(\mathbf{p}, \mathbf{q}) = \frac{\mathbf{p}^\intercal \mathbf{q}}{\|\mathbf{p}\|\|\mathbf{q}\|}$.

τ is the hyper-parameter and δ^c_k is the prior probability of the class c at the client k. This can be obtained by computing the empirical label distribution. Availability of such class prior information is a common assumption in the works such as Lee et al. (2022)

$$\delta^c_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \mathbb{I}_{y_{i,k}=c} \tag{14}$$

The final scores $s^c_{i,l-1,k}$ are obtained as

$$s^c_{i,l-1,k} = \frac{\hat{s}^c_{i,l-1,k}}{\sum_{m=1}^{|C|} \hat{s}^m_{i,l-1,k}} \tag{15}$$

The scores $s^c_{i,l-1,k}$ can be interpreted as the probability assigned to the class-specific prompt $\mathbf{p}_c$, given the `cls` token $\mathbf{cls}_{l-1,i,k}$. All the probabilities across the classes form the desired weight vector $\mathbf{s}_{i,l-1,k}$ and the final CCMP $\mathbf{m}_{l-1}$ is computed using Eq. 6. CCMP achieves personalization even when client priors are uniform, due to its ability to capture the domain gap within the scores. This is a direct result of the $\mathbf{cls}_{l-1,i,k}$ tokens, as they will have different representations for different domains.

Privacy considerations: We acknowledge that PEP-FedPT requires the transmission of class prototypes to facilitate adaptive prompt mixing. However, these transmitted statistics represent aggregated, intermediate layers', low-dimensional summaries of the local data rather than raw data. Moreover, without formal privacy protection measures, strict privacy guarantees are challenging in virtually any FL framework. This is discussed clearly in the Sec A.6 of (Xu et al.). Prototype-based learning in FL has been used previously in (Dai et al., 2023; Tan et al., 2022; Xu et al.). Additionally, in practice, while sharing the model information, client devices use a layer of privacy-preserving techniques such as differential privacy to further enhance the client model privacy. In A.4.1, we have discussed the impact of adding differential privacy noise to the class prototypes on the overall performance of our method. We have observed that it has had minimal impact on the final accuracy of our proposed method.

4.3 Theoretical Underpinnings of CCMP

4.3.1 CCMP minimizes the Quadratic Upper bound on the Loss around the class prompts

CCMP constitutes the core component of our methodology and by design, it is specific to each client. We show that CCMP minimizes the quadratic upper bound on the loss. We denote the estimate of the class prompt for class i in any round as $\mathbf{p}_{c_i}$ and the class prompts will be denoted by $\mathbf{P}_C = [\mathbf{p}_{c_1}, \mathbf{p}_{c_2} \dots, \mathbf{p}_{c_{|C|}}]$. Let $\mathbf{m}(k)$ denote the CCMP prompt used for client k [2], and let the total number of clients be n, and δ^i_k denote the empirical probability that a data point at client k belongs to class i. Let $\mathcal{P}$ be the set of all possible prompts across all the clients, such that $\mathbf{m}(k) \in \mathcal{P}, \quad \forall k \in \{1, 2, \dots, n\}$. We denote the average loss corresponding to a data point whose true label $y = i$ as l^i. We rewrite this loss in terms of prompt $\mathbf{p}$ and class i as $l^i(\mathbf{p})$. Then the global loss across all clients can be computed as $f := \frac{1}{n}\sum_{k=1}^{n}\left[\sum_{i=1}^{|C|}\delta^i_k \cdot l^i_k(\mathbf{m}(k))\right]$. We now state the following assumptions:

A 1. *$\mathcal{P}$ is compact subset of $\mathbb{R}^d$, where d is the token dimension.*

A 2. *l^i_k is Lipschitz smooth with parameter β_i $\forall i \in [|C|]$,$\forall k \in [n]$.*

A 3. *$l^i_k(\mathbf{p})$ achieves its' minimum value for $\mathbf{p} = \mathbf{p}^*_{c_i}$.*

Proposition 1. *If the above assumptions hold, we show that f can be upper bounded as $f \leq \tilde{L} = \frac{1}{n}\sum_{k=1,i=1}^{n,|C|}\delta^i_k\left(l^i_k(\mathbf{p}_{c_i}) + \frac{\beta_{\max}}{2}\|\mathbf{m}(k) - \mathbf{p}_{c_i}\|^2\right) + \tilde{C}$ and it is minimized at $\mathbf{m}(k) = \sum_{i=1}^{|C|}\delta^i_k\mathbf{p}_{c_i}, \quad \forall k \in [n]$. which is equivalent to the (CCMP) described in sec.4.2 as $\tau >> 1$. $\beta_{\max} = \max_{i\in[|C|]}\beta_i$, $\tilde{C}$ is a constant which depends on $\mathcal{P}$. This vanishes when $\mathbf{p}_{c_i} = \mathbf{p}^*_{c_i}\forall i \in [|C|]$ which makes $\tilde{L}$ a tight upper bound of f.*

A detailed proof is provided in Section A.6.1 of the appendix. The proof sketch proceeds by constructing an upper bound on the class-wise loss using smoothness assumptions. Specifically, the first-order term is bounded via compactness and smoothness properties. The upper bound is then minimized with respect to each $\mathbf{m}(k)$. The key insight from this proposition is that each client's prompt differs due to client-specific class priors, even though the underlying class prompts $\mathbf{p}_{c_i}$ are shared globally. This mechanism enables the prompts to adapt to each client's data distribution. The mixing scores in Eq. 15 depend both on the data instance and the class priors; this is further discussed in Sec A.6.4 of appendix.

4.3.2 CCMP is Optimal in Minimum Mean Squared Estimate (MMSE) Sense

We denote $p_k(\mathbf{p}|\mathbf{cls}_{l-1})$ as the posterior probability of the prompt $\mathbf{p}$ after observing the `cls` token $\mathbf{cls}_{l-1}$ at the input of layer l. [3]. It should be noted that this is a discrete probability measure over the class-specific prompts $\{\mathbf{p}_{c_1}, \mathbf{p}_{c_2} \dots, \mathbf{p}_{c_{|C|}}\}$. If we assume that the density over the `cls` tokens follows $p_k(\mathbf{cls}_{l-1})$ and the posterior over the class given the `cls` token is modeled as (based on Eq. 15) i.e.,

[2] for notation convenience, we drop the layer index j from $\mathbf{m}_j(k)$.

[3] In $\mathbf{cls}_{l-1}$ we omit the subscripts of client k, data point i and round t for simplifying notation.

$$p_k(\mathbf{p} = \mathbf{p}_c|\mathbf{cls}_{l-1}) = s^c_{i,l-1,k} \tag{16}$$

This induces the joint probability density over the `cls` tokens observed and the class-specific prompts $\{\mathbf{p}_{c_1}, \mathbf{p}_{c_2} \ldots, \mathbf{p}_{c_{|C|}}\}$. We denote this by $p_k(\mathbf{cls}_{l-1}, \mathbf{p})$ and is given below

$$p_k(\mathbf{cls}_{l-1}, \mathbf{p}) = p_k(\mathbf{p}|\mathbf{cls}_{l-1})p_k(\mathbf{cls}_{l-1}) \tag{17}$$

Proposition 2. *If the cls tokens and the class-specific prompts at input of layer l has the joint density given by $p_k(\mathbf{cls}_{l-1}, \mathbf{p})$ as in Eq. 17, then the CCMP prompt for a client k, $\mathbf{m}_{l-1}(k)$ obtained in Eq. 6 is Minimum Mean Squared Estimator (MMSE) of the true class prompt.*

The proposition 2 says that the CCMP obtained in Eq. 6 is optimal in MMSE sense. The detailed proof is given in Sec. A.6.2 of the appendix. This is done by showing MMSE optimality of the estimator $\mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}]$. The discussion regarding the convergence is provided in Sec. A.6.3 of the appendix.

5 Experiments

Datasets:

We conducted extensive experiments on four popular datasets (1) *CIFAR-100* (Krizhevsky & Hinton, 2009) dataset consists of 50,000 training images and $10,000$ test images distributed across 100 classes. (2) *TinyImageNet* (Le & Yang, 2015) contains $100K$ images of 200 classes, with each class containing 500 training images and 50 test images. (3) *DomainNet* (Peng et al., 2019) comprises 0.6 million images of 345 classes distributed across six domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch; however, following the protocol of Yang et al. (2023), we use the top ten most frequent classes for our experiments. (4) *iNaturalist* originally introduced in Van Horn et al. (2018) is a large-scale fine-grained visual classification dataset comprised of images of natural species. We use the federated version of this dataset introduced in the paper.

Setup:

We split CIFAR-100 and Tiny-ImageNet datasets among the clients with two different settings of data heterogeneity: pathological splitting (Li et al., 2023; Oh et al.; Deng et al., 2024), where each client observes only 10 classes, and Dirichlet-based splitting denoted by $Dir(\beta)$ (Acar et al.), where each client has a non-identical label distribution. A lower β value indicates higher heterogeneity, and we set $\beta = 0.3$. For CIFAR-100 we consider 100 clients and for TinyImageNet we consider 200 clients. Only 5 randomly chosen clients participate in every round. For the DomainNet dataset we consider the setting as Deng et al. (2024); Li et al. (2021) where each domain is allocated to 10 clients among 60 clients, this considers the scenario of feature imbalance setting. 6 randomly sampled clients participate in each communication round. Finally, in the iNaturalist dataset (Hsu et al., 2020), we make sure each client gets at least 16 training samples. This will have around $100k$ training samples distributed among the 1018 clients and 1203 classes. The partition is performed to mimic the cross-device (Kairouz et al., 2021) non-iid setting. In all the above experimental setups, the partition of the dataset across clients is completely disjoint, i.e., no two clients contain the same data sample in all cases. The visualization of data heterogeneity is provided in Sec. A.3.1 of appendix.

Model Details:

We use the Vision Transformer (ViT-B/16)(Neil & Dirk, 2020) pre-trained on the ImageNet-21K dataset (Feng et al., 2023; Yang et al., 2023) as our base model. ViT-B-16 was originally trained on images with a resolution of 224x224 pixels, utilizing a patch size of 16. To maintain compatibility, we resize our input images to 224x224 pixels. In the prompt-tuning stage, we specifically focus on optimizing shared and class prompts, as well as the classifier head. The hyper-parameter details are in Sec. A.3.2 of the appendix.

Baselines:

We compared our method against several global and personalized Federated Learning (FL) methods that use the Prompt Tuning including Head-Tuning(Sun et al., 2022), FedVPT and FedVPT-D (Jia et al., 2022),

Table 1: Quantitative comparisons on CIFAR-100, Tiny-ImageNet datasets using ViT-B/16. We report the accuracy under two non-iid data partitioning setups 1) pathological: Each client observes only 10 classes 2) Dirichlet: Label distribution of each client is drawn from Dirichlet distribution.

Datasets	CIFAR-100 (%) ↑				Tiny-ImageNet (%) ↑			
Method	Pathological		$Dir(0.3)$		Pathological		$Dir(0.3)$	
	Mean Acc	Worst Acc	Mean Acc	Worst Acc	Mean Acc	Worst Acc	Mean Acc	Worst Acc
Head-Tuning	77.85$_{\pm 0.17}$	59.87$_{\pm 0.74}$	79.56$_{\pm 0.25}$	66.66$_{\pm 2.79}$	68.39$_{\pm 0.76}$	44.09$_{\pm 0.58}$	70.73$_{\pm 0.08}$	45.63$_{\pm 2.05}$
FedVPT	83.62$_{\pm 0.02}$	70.19$_{\pm 0.11}$	84.91$_{\pm 0.07}$	74.64$_{\pm 0.74}$	74.20$_{\pm 0.33}$	54.00$_{\pm 2.46}$	76.57$_{\pm 0.34}$	50.34$_{\pm 2.51}$
FedVPT-D	85.15$_{\pm 0.77}$	70.12$_{\pm 0.20}$	88.60$_{\pm 0.19}$	79.17$_{\pm 0.65}$	79.60$_{\pm 0.42}$	59.83$_{\pm 1.66}$	83.30$_{\pm 0.16}$	60.33$_{\pm 0.58}$
FedPR	81.77$_{\pm 0.30}$	68.99$_{\pm 0.48}$	82.27$_{\pm 0.22}$	73.29$_{\pm 1.38}$	68.86$_{\pm 0.17}$	47.50$_{\pm 1.63}$	68.93$_{\pm 0.11}$	47.37$_{\pm 1.44}$
SGPT	84.16$_{\pm 0.24}$	70.79$_{\pm 0.30}$	85.90$_{\pm 0.21}$	76.73$_{\pm 1.60}$	75.65$_{\pm 1.81}$	55.66$_{\pm 3.32}$	78.84$_{\pm 1.11}$	53.87$_{\pm 0.46}$
pFedPG	92.96$_{\pm 1.34}$	84.58$_{\pm 1.1}$	77.27$_{\pm 0.77}$	62.34$_{\pm 1.53}$	82.93$_{\pm 0.18}$	50.21$_{\pm 1.05}$	55.91$_{\pm 0.65}$	49.31$_{\pm 1.05}$
P-PT	75.37$_{\pm 0.39}$	55.14$_{\pm 1.03}$	80.10$_{\pm 0.25}$	68.33$_{\pm 0.58}$	61.68$_{\pm 1.16}$	38.09$_{\pm 6.38}$	62.78$_{\pm 0.38}$	40.30$_{\pm 1.13}$
PEP-FedPT(Ours)	**95.46**$_{\pm 0.16}$	**84.74**$_{\pm 3.12}$	**88.75**$_{\pm 0.25}$	**81.00**$_{\pm 0.00}$	**91.52**$_{\pm 0.11}$	**77.33**$_{\pm 1.84}$	**83.44**$_{\pm 0.02}$	**61.00**$_{\pm 0.31}$

FedPR (Feng et al., 2023), and SGPT (Deng et al., 2024). To thoroughly assess our method's performance, we introduce a new baseline called P-PT which personalizes the prompts, giving insights into how the personalization of prompts impacts performance.

Evaluation Methodology:

We use two key metrics to assess the performance of both the baselines and our proposed method (Deng et al., 2024): (1) *Mean Accuracy* calculates the average accuracy across individual clients' test data, reflecting adaptation to diverse client data distributions. (2) *Worst Local Accuracy* reflects the performance of the worst-performing client, indicating adaptation to the most challenging local data. Furthermore, we utilize a *heldout evaluation* strategy (Yuan et al., 2021), where 90% of clients participate in training and 10% are reserved for testing. All aforementioned metrics are reported separately for both participating and heldout clients. This setting demonstrates the model's effectiveness in onboarding new clients and adapting to previously unseen data without sharing updates with the central server. All our experiments are performed over 3 different initializations and the mean and standard deviations are reported as ($mean_{\pm std}$).

5.1 Results and Discussion

5.1.1 Label Heterogeneity Results

The class heterogeneity results are presented in Table- 1. PEP-FedPT outperforms all the baselines, e.g., when compared against pFedPG with CIFAR-100 pathological setting we observe an improvement of 2.57% in mean accuracy. For TinyImagenet the improvement is 8.59% (mean accuracy) over pFedPG and 17.5% (worst accuracy) over FedVPT-D. This shows that our prompt-mixing mechanism effectively addresses data heterogeneity independently and performs even better in scenarios of higher heterogeneity or lower class overlap between clients. Similar improvements are seen in other settings. Additional experiments are provided in Sec. A.4 of the appendix. The visualization of the accuracy vs communication rounds is shown in Sec. A.5.1.

5.1.2 Feature Heterogeneity Results

Feature-heterogeneity results are presented in Table- 2. FedVPT-D serves as strong baseline due to its inclusion of prompts at each layer. We present the results for our method. In this setting, personalized method pFedPG also proves highly beneficial due to the pronounced feature imbalance among clients. Our method PEP-FedPT improves on average by 5.52% over the best performing baseline FedVPT-D on the iNaturalist dataset.

5.1.3 Heldout Evaluation

In the Table 3 we present the results for the heldout evaluation, where we report the accuracies of the participating clients and the new clients. Participating clients are the ones who participate in the federated

Table 2: Experimental results on DomainNet and iNaturalist. For DomainNet we consider each domain belonging to a client and we report the accuracy attained by each client and the average accuracy. On the iNaturalist dataset we report the average test accuracy of all the clients. Our method significantly outperforms all the baselines on this challenging dataset.

Datasets	DomainNet(%) ↑								iNaturalist(%) ↑	
Method	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Mean Acc	Worst Acc	Mean Acc	Worst Acc
Head-Tuning	$91.16_{\pm 0.92}$	$57.45_{\pm 1.27}$	$91.39_{\pm 0.24}$	$74.94_{\pm 0.28}$	$96.68_{\pm 0.42}$	$86.46_{\pm 1.42}$	$83.71_{\pm 1.27}$	$38.88_{\pm 3.70}$	$49.41_{\pm 0.41}$	$46.52_{\pm 1.82}$
FedVPT	$90.84_{\pm 1.37}$	$58.56_{\pm 0.60}$	$92.25_{\pm 0.67}$	$77.81_{\pm 0.30}$	$96.78_{\pm 0.48}$	$88.24_{\pm 0.89}$	$84.23_{\pm 0.72}$	$37.02_{\pm 2.44}$	$52.22_{\pm 0.50}$	$50.19_{\pm 0.52}$
FedVPT-D	$94.01_{\pm 1.05}$	$63.29_{\pm 0.81}$	$\mathbf{93.45}_{\pm 0.49}$	$84.56_{\pm 1.59}$	$96.96_{\pm 0.66}$	$91.58_{\pm 0.09}$	$87.31_{\pm 0.51}$	$42.49_{\pm 1.85}$	$57.96_{\pm 1.12}$	$53.61_{\pm 1.22}$
FedPR	$91.62_{\pm 0.91}$	$56.20_{\pm 1.01}$	$91.16_{\pm 1.17}$	$73.72_{\pm 0.62}$	$96.66_{\pm 0.39}$	$86.41_{\pm 1.01}$	$82.95_{\pm 1.26}$	$35.18_{\pm 3.70}$	$41.25_{\pm 2.31}$	$25.40_{\pm 1.58}$
SGPT	$92.64_{\pm 0.65}$	$60.62_{\pm 0.22}$	$91.54_{\pm 0.78}$	$83.55_{\pm 1.85}$	$96.55_{\pm 0.17}$	$89.93_{\pm 0.47}$	$85.56_{\pm 0.60}$	$37.34_{\pm 1.41}$	$55.78_{\pm 0.57}$	$45.41_{\pm 1.02}$
pFedPG	$92.89_{\pm 0.82}$	$63.56_{\pm 0.88}$	$92.27_{\pm 1.01}$	$\mathbf{87.33}_{\pm 0.21}$	$97.16_{\pm 0.25}$	$89.34_{\pm 0.30}$	$87.40_{\pm 0.30}$	$52.05_{\pm 0.32}$	$52.42_{\pm 2.59}$	$39.58_{\pm 0.49}$
P-PT	$90.11_{\pm 1.41}$	$56.73_{\pm 1.16}$	$90.25_{\pm 1.09}$	$74.81_{\pm 0.81}$	$95.18_{\pm 0.70}$	$85.26_{\pm 0.53}$	$82.30_{\pm 1.31}$	$35.67_{\pm 1.33}$	$45.69_{\pm 0.52}$	$40.09_{\pm 0.68}$
PEP-FedPT(Ours)	$\mathbf{95.46}_{\pm 0.41}$	$\mathbf{71.68}_{\pm 1.41}$	$93.00_{\pm 0.55}$	$86.89_{\pm 1.53}$	$\mathbf{97.67}_{\pm 0.53}$	$\mathbf{91.79}_{\pm 0.79}$	$\mathbf{89.15}_{\pm 0.70}$	$\mathbf{59.79}_{\pm 2.52}$	$\mathbf{63.48}_{\pm 1.10}$	$\mathbf{57.61}_{\pm 0.82}$

training and the new clients do not participate in the FL training. We consider the CIFAR-100 dataset with pathological partitioning where each client observes only 10 classes. Most baseline methods achieve competitive accuracy on participating clients but show reduced performance on unseen clients. Personalized methods like pFedPG achieve very high participating accuracy but fail in held-out testing, highlighting poor generalization. In contrast, our method consistently achieves the best results across both datasets, with 95.66% vs. 93.71% on CIFAR-100 and 92.53% vs. 90.60% on Tiny-ImageNet, showing strong generalization to unseen clients. The results on DomainNet and iNaturalist are provided in Sec. A.4.7 of the appendix.

Table 3: Quantitative comparisons on CIFAR-100 and Tiny-ImageNet with held out evaluation: We report the accuracy with pathological partitioning where each client observes 10 classes. It can be observed that the personalized methods like pFedPG perform the worst in the held-out evaluation Our method performs well on the clients participating in the FL training and also on the unseen clients.

Method	CIFAR-100 (↑)		Tiny-ImageNet (↑)	
	Participating Acc	Testing Acc	Participating Acc	Testing Acc
Head	$77.81_{\pm 0.25}$	$77.10_{\pm 0.41}$	$67.97_{\pm 0.66}$	$68.97_{\pm 0.70}$
FedVPT	$83.62_{\pm 0.24}$	$82.39_{\pm 1.12}$	$74.15_{\pm 0.47}$	$74.15_{\pm 1.19}$
FedVPT-D	$85.06_{\pm 0.51}$	$84.87_{\pm 0.44}$	$77.38_{\pm 1.35}$	$76.89_{\pm 0.90}$
FedPR	$81.62_{\pm 0.27}$	$80.61_{\pm 0.68}$	$69.37_{\pm 1.42}$	$68.23_{\pm 1.90}$
SGPT	$83.90_{\pm 0.23}$	$83.63_{\pm 0.64}$	$76.38_{\pm 0.68}$	$78.10_{\pm 1.85}$
pFedPG	$93.32_{\pm 0.85}$	$08.35_{\pm 0.21}$	$86.09_{\pm 1.42}$	$04.81_{\pm .48}$
P-PT	$75.97_{\pm 1.38}$	$72.19_{\pm 0.58}$	$60.76_{\pm 0.39}$	$60.41_{\pm 1.40}$
PEP-FedPT(Ours)	$\mathbf{95.66}_{\pm 0.17}$	$\mathbf{93.71}_{\pm 0.40}$	$\mathbf{92.53}_{\pm 0.35}$	$\mathbf{90.60}_{\pm 0.61}$

5.2 Analysis of PEP-FedPT

5.2.1 Ablations

We conducted ablation studies on shared and class-specific prompts evaluating their impact by varying the number of shared prompts and the influence of class priors on the prompt mixing strategy. Table 4 reports the effect of combining shared prompts with CCMP under both Pathological and Dirichlet splits. For the Pathological split, the average accuracy improves from 83.62% with only shared prompts to 95.46% with shared + CCMP. Similarly, for the Dirichlet split, the performance increases from 84.91% to 88.75%. These results highlight the consistent benefit of incorporating CCMP across different data partitioning strategies.

The Table 5 presents the effect of incorporating class priors into the prompt design on CIFAR-100 under both Pathological and Dirichlet splits. The results show that using Shared + CCMP with Class Priors (CP) consistently improves performance over the baseline without CP. In particular, the Pathological split benefits, with accuracy increasing from 84.01% to 95.46%, while the Dirichlet split also shows a notable gain from 86.12% to 88.75%. Similar analysis for other datasets is given in section A.4.3 and A.4.2 of the appendix. Further additional experiments can be found in Section A.4 of the appendix.

Table 4: Shared and CCMP ablation on the CIFAR-100 dataset with Dirichlet and Pathological Partitions. We report the Mean Accuracy in (%)

Prompt Strategy	Pathological Split	Dirichlet Split
Only Shared	$83.62_{\pm 0.02}$	$84.91_{\pm 0.07}$
Shared + CCMP	$95.46_{\pm 0.16}$	$88.75_{\pm 0.25}$

Table 5: Impact of Class Priors on CIFAR-100 dataset with Dirichlet and Pathological Partitions

Prompt Strategy	Pathological Split	Dirichlet Split
Shared + CCMP Without CP	$84.01_{\pm 0.04}$	$86.12_{\pm 0.13}$
Shared + CCMP With CP	$95.46_{\pm 0.16}$	$88.75_{\pm 0.25}$

5.2.2 Computation and Communication

We denote that d and d_h are token and attention head dimensions, C and L denote the number of classes and layers respectively, and T are the tokens. The minimum computations required by ViT forward is given as: The Query (Q), Key (K) and Value (V) requires Tdd_h multiplications each. The inner product matrix QK^T requires T^2d_h multiplications. The feedforward computations requirement is d^2T. If H heads are present and there are C classes for the classification then we need $LH(3Tdd_h + T^2d_h) + LTd^2 + Cd$ multiplications. For CIFAR-100 on ViT-B/16 the CCMP computation takes only 0.008% of total computations, which is very negligible implying the efficiency of the CCMP computation.

Table 6: Comparison of computation and communication. We analyze the resources required to achieve the target accuracy of 83% on CIFAR-100 dataset

Method	**Training Time (sec)** ↓	**Params Communicated** ↓	**Rounds Required** ↓
FedVPT	4550	7.7 M	100
FedVPT-D	4760	7.67 M	90
SGPT	8170	13 M	90
FedPR	> 5360	> 8.4M	> 100
PEP-FedPT(Ours)	**1153**	**4.6** M	**12**

Table 6 compares the computational and communication complexity of our proposed method, PEP-FedPT against the different baselines. For a fair comparison, we compare the methods that use global prompts. We analyze the resources required to achieve 83% accuracy, which is the highest accuracy reported for FedVPT. Our results show that PEP-FedPT achieves this accuracy in just 12 rounds, requiring lowest training time and significantly reducing communication overhead (4.6M) compared to SGPT (13.0M), where M denotes million. The claim of 12 rounds can be verified in the Figure 3. FedPR only attains 81.66% in 100 rounds so we report this as (> 100). The training times reported are measured on an Nvidia RTX-A6000 GPU.
The detailed computation of why **4.6** M is : Head requires (100×768), shared prompt (1×768) class prompts (100×768), prototypes $(100 \times 768 \times 3)$ scaled by 3 because of three layers with CCMP prompts. Total rounds 12 and in total, yields 4.6M parameters. In this communication analysis, since our method shares the global prompts, we compared only the methods that are not personalized.

6 Limitations and Scope for Future Work

While our approach achieves strong performance and generalization, it relies on the empirical estimation of class priors and `cls` token centroids, both of which require access to labeled data on the client side. In scenarios such as semi-supervised or unsupervised federated learning, where labeled data is scarce or unavailable, these estimates may become unreliable or noisy, potentially degrading the quality of the constructed prompts. Consequently, adapting our method to these settings is non-trivial. Exploring such extensions presents a promising direction for future research.

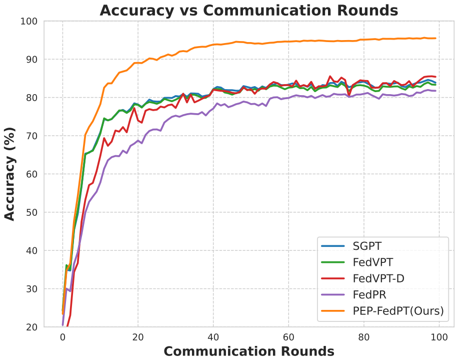


Figure 3: Comparison of the convergence of different methods across the Communication rounds on the CIFAR-100 dataset with pathological non-iid partitioning where each client only observes 10 classes.

7 Conclusion

We propose a novel prompt-tuning methodology for Vision Transformers (ViTs) by introducing class-specific prompts alongside shared prompts. Our approach leverages the `cls`-token representations in pretrained ViT layers to extract prototypes, which are then combined with each client's prior label distribution to compute soft scores that guide the mixing of class-specific prompts into a unified, optimized prompt (CCMP). This dynamic mixing allows (CCMP) to achieve personalization while using global prompts only. This combined prompt (CCMP) is subsequently embedded within the ViT layer. Our method PEP-FedPT, achieves State of the Art performance, surpassing previous methods across the benchmark datasets.

References

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 112:59–67, 2018.

Debora Caldarola, Barbara Caputo, and Marco Ciccone. Improving generalization in federated learning by seeking flat minima. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, pp. 654–672. Springer, 2022.

Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *International Conference on Learning Representations*.

Yutong Dai, Zeyuan Chen, Junnan Li, Shelby Heinecke, Lichao Sun, and Ran Xu. Tackling data heterogeneity in federated learning with class prototypes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7314–7322, 2023.

W. Deng, C. Thrampoulidis, and X. Li. Unlocking the potential of prompt-tuning in bridging generalized and personalized federated learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6087–6097. IEEE Computer Society, 2024. doi: 10.1109/CVPR52733.2024.00582. URL `https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.00582`.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

Chun-Mei Feng, Bangjun Li, Xinxing Xu, Yong Liu, Huazhu Fu, and Wangmeng Zuo. Learning federated visual prompt in null space for mri reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8064–8073, 2023.

Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10112–10121, June 2022.

Andrew Hard, Chloé M Kiddon, Daniel Ramage, Francoise Beaufays, Hubert Eichner, Kanishka Rao, Rajiv Mathews, and Sean Augenstein. Federated learning for mobile keyboard prediction, 2018. URL `https://arxiv.org/abs/1811.03604`.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pp. 76–92. Springer, 2020.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pp. 709–727. Springer, 2022.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Canadian Institute for Advanced Research, 2009.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. Preservation of the global knowledge by not-true distillation in federated learning. *Advances in Neural Information Processing Systems*, 35:38461–38474, 2022.

Hongxia Li, Zhongyi Cai, Jingya Wang, Jiangnan Tang, Weiping Ding, Chin-Teng Lin, and Ye Shi. Fedtp: Federated learning by transformer personalization. *IEEE transactions on neural networks and learning systems*, 2023.

Hongxia Li, Wei Huang, Jingya Wang, and Ye Shi. Global and local prompts cooperation via optimal transport for federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12151–12161, 2024.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL `https://aclanthology.org/2021.acl-long.353`.

Xiaoxiao Li, Meirui JIANG, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-IID features via local batch normalization. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=6YEQUn0QICG`.

Chih-Ting Liu, Chien-Yi Wang, Shao-Yi Chien, and Shang-Hong Lai. Fedfr: Joint optimization federated framework for generic and personalized face recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 1656–1664, 2022a.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, 2022b.

Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10082–10091, 2022. doi: 10.1109/CVPR52688.2022.00985.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Jed Mills, Jia Hu, and Geyong Min. Communication-efficient federated learning for wireless edge intelligence in iot. *IEEE Internet of Things Journal*, 7(7):5986–5994, 2019.

Houlsby Neil and Weissenborn Dirk. Transformers for image recognition at scale. *Online: https://ai. googleblog. com/2020/12/transformers-for-image-recognitionat. html*, 2020.

Dinh C Nguyen, Ming Ding, Pubudu N Pathirana, Aruna Seneviratne, Jun Li, and H Vincent Poor. Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23 (3):1622–1658, 2021.

Jaehoon Oh, SangMook Kim, and Se-Young Yun. Fedbabu: Toward enhanced representation for federated image classification. In *International Conference on Learning Representations*.

Oleksiy Ostapenko, Timothee Lesort, Pau Rodriguez, Md Rifat Arefin, Arthur Douillard, Irina Rish, and Laurent Charlin. Continual learning with foundation models: An empirical study of latent replay. In Sarath Chandar, Razvan Pascanu, and Doina Precup (eds.), *Proceedings of The 1st Conference on Lifelong Learning Agents*, volume 199 of *Proceedings of Machine Learning Research*, pp. 60–91. PMLR, 22–24 Aug 2022.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.

Liangqiong Qu, Yuyin Zhou, Paul Pu Liang, Yingda Xia, Feifei Wang, Ehsan Adeli, Li Fei-Fei, and Daniel Rubin. Rethinking architecture design for tackling data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10061–10071, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. Federated learning for emoji prediction in a mobile keyboard. *arXiv preprint arXiv:1906.04329*, 2019.

Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020.

Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pp. 9489–9502. PMLR, 2021.

Guangyu Sun, Matias Mendieta, Taojiannan Yang, and Chen Chen. Exploring parameter-efficient fine-tuning for improving communication efficiency in federated learning. 2022.

Yan Sun, Li Shen, Tiansheng Huang, Liang Ding, and Dacheng Tao. Fedspeed: Larger local interval, less communication round, and higher generalization accuracy. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=bZjxxYURKT`.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):9587–9603, 2023. doi: 10.1109/TNNLS.2022.3160699.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8432–8440, 2022.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

Shanshan Wu, Tian Li, Zachary Charles, Yu Xiao, Ken Liu, Zheng Xu, and Virginia Smith. Motley: Benchmarking heterogeneity and personalization in federated learning. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*.

Jian Xu, Xinyi Tong, and Shao-Lun Huang. Personalized federated learning with feature alignment and classifier collaboration. In *The Eleventh International Conference on Learning Representations*.

Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5:1–19, 2021.

Fu-En Yang, Chien-Yi Wang, and Yu-Chiang Frank Wang. Efficient model personalization in federated learning via client-specific prompt generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19159–19168, 2023.

Honglin Yuan, Warren Morningstar, Lin Ning, and Karan Singhal. What do we mean by generalization in federated learning? *arXiv preprint arXiv:2110.14216*, 2021.

Weiming Zhuang, Yonggang Wen, Xuesen Zhang, Xin Gan, Daiying Yin, Dongzhan Zhou, Shuai Zhang, and Shuai Yi. Performance optimization of federated person re-identification via benchmark analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 955–963, 2020.

A Appendix

A.1 Overview of Notation and Definitions

We give the overview of the notations and definitions used in the paper

- Boldface letters to denote matrices and vectors.
- "$\cdot$" refers to element-wise multiplication.
- $*$ refers to the standard matrix multiplication.
- We define TL_i as the i^{th} transformer layer.
- $\mathbb{E}$ denotes the expectation operator.

- The term δ_k^c represents the prior probability of class c occurring at client k.
- Additionally, $sim(\mathbf{p}, \mathbf{q})$ denotes the cosine similarity, while $\mathbb{I}$ represents the indicator function.
- $\boldsymbol{\mu}_{l-1,k,t}^c$ denotes the client k computing the CLS token prototypes in communication round t and the input of layer l.
- $\mathbf{H}$ denotes the classification head.
- $\|.\|$ denotes Euclidean norm or 2-norm.
- $[M]$ denotes the set $\{1, 2, ., .M\}$.
- $\Lambda = \{rR, rR+1, \dots, (r+1)R-1\}$ denotes the set of communication rounds in r-th update period of length R.
- $n_{k,c}$ denotes the number of samples corresponding to class c in client k.
- $\mathbf{P_S}$ and $\mathbf{P_C}$ denote shared prompts and class-specific prompts respectively.
- $\mathbf{m}$ denotes the CCMP.
- N_k denotes the total number of datapoints at client k and n denotes the number of clients.

A.2 Method Details: Algorithm

We briefly go over the prototype update and the CCMP computation equations. Client level prototype aggregation at communication round t is given in the below Eq. 18

$$\boldsymbol{\mu}_{l-1,k,t}^c = \begin{cases} \frac{1}{n_{k,c}} \sum_{i=1}^{N_k} \mathbf{cls}_{l-1,i,k,t} \cdot \mathbb{I}_{y_{i,k}=c}, & n_{k,c} > 0, \\ \mathbf{0}, & n_{k,c} = 0. \end{cases} \tag{18}$$

Server prototype aggregation during the warm-up phase is given by Eq. 19

$$\boldsymbol{\mu}_{l-1,0}^c = \frac{1}{|S_0|} \sum_{k \in S_0} \boldsymbol{\mu}_{l-1,k,0}^c \tag{19}$$

Server aggregation of the prototypes at the end of r-th update period is in Eq. 20

$$\hat{\boldsymbol{\mu}}_{l-1,r}^c = \begin{cases} \frac{1}{D_c} \sum_{t \in \Lambda} \sum_{k \in S_t} \mathbb{I}_{\{\boldsymbol{\mu}_{l-1,k,t}^c \neq \mathbf{0}\}} \, \boldsymbol{\mu}_{l-1,k,t}^c, & D_c > 0, \\ \mathbf{0}, & D_c = 0. \end{cases} \tag{20}$$

Sever updating the prototypes based on the momentum is given in Eq. 21. If D_c is 0, we set $\rho = 1$.

$$\boldsymbol{\mu}_{l-1,r}^c = \rho \cdot \boldsymbol{\mu}_{l-1,r-1}^c + (1-\rho) \cdot \hat{\boldsymbol{\mu}}_{l-1,r}^c \tag{21}$$

The class prior for class c at client k is computed as in Eq. 22

$$\delta_k^c = \frac{1}{N_k} \sum_{i=1}^{N_k} \mathbb{I}_{y_{i,k}=c} \tag{22}$$

The soft scores are computed based on similarity between the class prototypes and cls representations as in Eq. 23

$$\hat{s}^{c}_{i,l-1,k} = exp\left(\frac{sim\left(\mathbf{cls}_{l-1,i,k}, \boldsymbol{\mu}^{c}_{l-1}\right)}{\tau}\right)\delta^{c}_{k} \tag{23}$$

The scores are converted to probabilities using Eq. 24

$$s^{c}_{i,l-1,k} = \frac{\hat{s}^{c}_{i,l-1,k}}{\sum_{j=1}^{|C|}\hat{s}^{j}_{i,l-1,k}} \tag{24}$$

The probabilities serve as weights of class-specific prompts which produce the Class Contexualized Mixed Prompts (CCMP) as in Eq. 25

$$\mathbf{m}_{l-1} = \mathbf{P}_{C} * \mathbf{s}_{i,l-1,k} \tag{25}$$

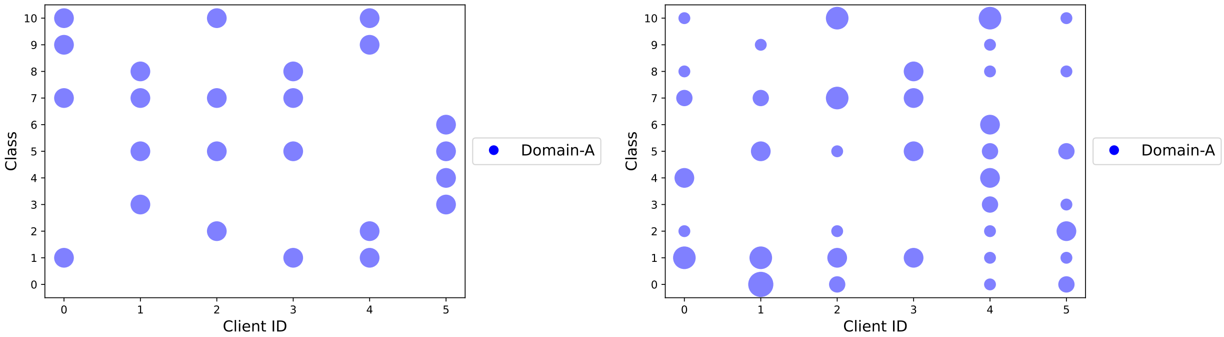
$\mathbf{s}_{i,l-1,k}$ is the vector containing $s^{c}_{i,l-1,k}$ for different values of c.

A.3 Experimental Setup

A.3.1 Details on Heterogeneity

We consider two different kinds of heterogeneity label imbalance and feature imbalance. In the label imbalance we again consider two different settings, pathological and the Dirichlet based non-iid settings as shown in Figure 4.

For pathological settings we select few classes of data points for each client and allocate the data among those labels. For Dirichlet we allocate the data by drawing a sample from the Dirichlet distribution. We consider these settings using the CIFAR-100 and Tiny-ImageNet Datasets by distributing the data among the 100 and 200 clients respectively and sampling only 5 clients in each communication round. For Dirichlert settings the degree of non-iid is controlled by the parameter δ and its denoted by $Dir(\delta)$. The lower delta implies higher heterogeneity and higher value implies the lower heterogeneity. Throughout the work we consider the value of δ to be 0.3.



(a) Non-IID of Label Shift: Pathological

(b) Non-IID of Label Shift: Dirichlet

Figure 4: Comparison of Non-IID Label Shift due to Pathological setting and the Dirichlet setting

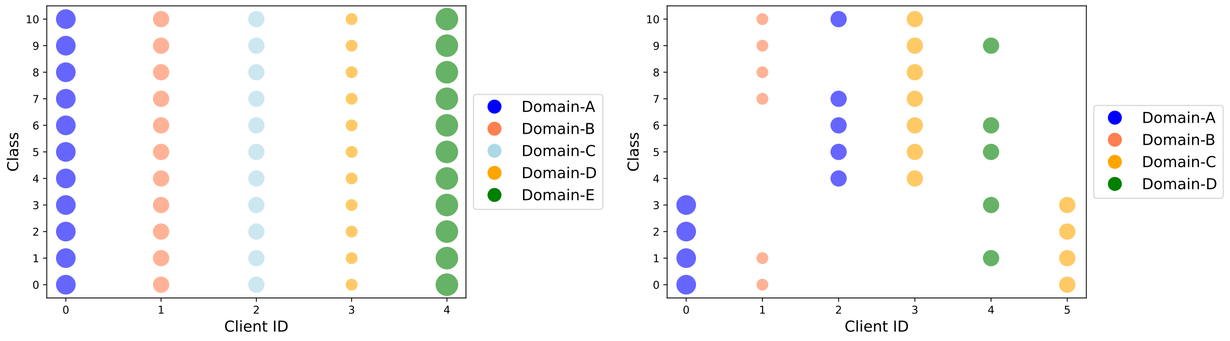
By feature imbalance, we mean clients are distributed with different domains. It can be seen in the Figure. 5. The DomainNet dataset can be viewed as analogous to the one described in the Figure 5a. In the Figure 5b the split shows the mix of feature and the label imbalance.

A.3.2 HyperParameter Details

We set the communication rounds to be 100 for CIFAR-100 and Tiny-ImageNet datasets. For DomainNet we set the rounds to 50. We set the number of Epochs to 5 for all the experiments except the iNaturalist, which

Algorithm 1: PEP-FedPT

```
Input: $\mathbf{H}$, $\mathbf{P_S}$,$\mathbf{P_C}$ $\mu$, Pretrained Vision Transformer $\mathbf{w}_{pre}$,Training data $(x, y) \sim \mathcal{D}$, Set of class labels $C$,
       Learning rate $\eta$, Number of local epochs $E$, Update period $R$, Total number of communication
       rounds $T$, Total number of clients $n$, CCMP layer index $l$
Output: $\boldsymbol{\theta} = \{\mathbf{H}, \mathbf{P_S}, \mathbf{P_C}\}$, $\boldsymbol{\mu}_{l-1}$
Server samples the subset $S_0 \subset [n]$
$\boldsymbol{\mu}^c_{l-1} \leftarrow WarmStartUp(\mathbf{w}_{pre}, S_0)\ \forall c \in C$
for  round $t \in [T]$ do
    Server samples participating clients $S_t \subset [n]$
    for client $k \in [S_t]$ do
        $\boldsymbol{\theta}^t_k$, $\boldsymbol{\mu}^c_{l-1,k,t}$ = LocalTrain $(\boldsymbol{\theta}^t, \boldsymbol{\mu}_{l-1}, l, k, t)$; $\forall c \in C$
$\boldsymbol{\theta}^{t+1} \leftarrow FedAveraging(\{\boldsymbol{\theta}^t_k, k \in S_t\})$McMahan et al. (2017)
if  $t$  mod $R = 0$ then
    $r \leftarrow \frac{t}{R}$
    for $c \in C$ do
        $\hat{\boldsymbol{\mu}}^c_{l-1,r} \leftarrow AggregateCentroids(\{\boldsymbol{\mu}^c_{l-1,k,t}, t \in \Lambda\})$ [Eq. 20]
        $\boldsymbol{\mu}^c_{l-1,r} \leftarrow UpdateCentroids(\hat{\boldsymbol{\mu}}^c_{l-1,r}, \boldsymbol{\mu}^c_{l-1,r-1})$; [Eq.21]
        $\boldsymbol{\mu}^c_{l-1} \leftarrow \boldsymbol{\mu}^c_{l-1,r}$
Return $\boldsymbol{\theta}, \boldsymbol{\mu}_{l-1}$
Function WarmStartUp($\mathbf{w}_{pre}$,$S_{in}$,$l$):
    for client $k$ in $S_{in}$ do
        obtain $\boldsymbol{\mu}^c_{l-1,k,0}$ [Eq. 10]
        Return $\boldsymbol{\mu}^c_{l-1,k,0} \forall c \in C$
    Server obtains $\boldsymbol{\mu}^c_{l-1,0}$ [Eq. 19]
    Return $\boldsymbol{\mu}^c_{l-1,0}$
Function LocalTrain($\boldsymbol{\theta}^t, \boldsymbol{\mu}_{l-1}, l, k, t$):
    compute $\boldsymbol{\mu}^c_{l-1,k,t}$ $\forall c \in C$ Eq. 10
    $\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}^t$
    for $e = 1 \rightarrow E$ do
        $\mathbf{m} \leftarrow PromptMixing((\mathbf{H}, \mathbf{P_S}, \mathbf{P_C}, \mathbf{w}_{pre}, \boldsymbol{\mu}_{l-1})$ [Eq. 22, 23, 24, 25]
        Define loss $l = l(\mathbf{H}, \mathbf{P_S}, \mathbf{m}, x, y)$
        $\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_k - \eta \cdot \nabla l_{\boldsymbol{\theta}_k}$
    Return $\boldsymbol{\theta}_k$, $\boldsymbol{\mu}^c_{l-1,k,t}$
```



(a) Feature Imbalance

(b) Feature Imbalance along with label shift

Figure 5: Comparison of Non-IID Feature Shift

is set to 2. The total communication rounds is set to 500 for iNaturalist. We follow stochastic Gradient Descent with momentum (Deng et al., 2024) as the default optimizer with learning rate 0.1 with exponential decay and the momentum 0.9. For the various datasets used in our experiments, we adapt the number of training rounds accordingly: 100 rounds for CIFAR-100 and Tiny-ImageNet, 50 rounds for DomainNet, and 500 rounds for iNaturalist. For all the experiments we consider number of shared prompts (n_S) to be 1, unless explicitly mentioned. We add the class specific prompts at the layers 5, 6 and 7. We also set the gradient clipping to 10 following Acar et al.. For all our experiment we consider number of shared prompts to 1 except the Tiny-ImageNet Dirichlet where we set it to 5. The CCMP is inserted at the layers 5, 6 and 7. We set the the temperature parameter τ to 0.05 for all our experiments.[4].

A.4 Additional Experiments

A.4.1 Class-Level Differential Privacy via Laplace Mechanism

Table 7: Impact of class-level DP noise ($\epsilon = 0.2$) on mean accuracy across datasets.

Method	CIFAR-100 (Path)	CIFAR-100 (Dir-0.3)	Tiny-ImageNet (Path)	Tiny-ImageNet (Dir-0.3)
With DP Noise	$93.23_{\pm 0.07}$	$86.92_{\pm 0.07}$	$91.16_{\pm 0.13}$	$82.92_{\pm 0.11}$
Without DP Noise	$95.46_{\pm 0.16}$	$88.75_{\pm 0.25}$	$91.52_{\pm 0.11}$	$83.44_{\pm 0.02}$

To protect individual class privacy in a federated learning setting, we employ the most common Laplace mechanism as described in Dwork et al. (2006) for class prototype during each server update. After aggregating class-specific model parameters from clients, we estimate the sensitivity of each class c based on the maximum L1 deviation of its CLS- token representation from the corresponding class prototype, normalized by the number of samples N_c:

$$S_c = \frac{2 \cdot \max_i \|\mathbf{cls}_i^{(c)} - \mu_c\|_1}{N_c}$$

where $\mathbf{cls}_i^{(c)}$ denotes the CLS token of the i-th sample belonging to class c, and μ_c is the prototype representing class c in the embedding space. To enforce differential privacy, Laplace noise is added to each centroid θ_c based on its sensitivity S_c and a predefined privacy budget ϵ:

$$\theta_c \leftarrow \theta_c + \text{Laplace}(0, S_c/\epsilon)$$

This class-aware noise injection is performed at every server update, for all the CCMP layers. As a result, individual class-level contributions are obfuscated, thereby enhancing privacy while preserving model performance under non-IID data distributions. In the Table 7 we have shown the impact of dp noise on our overall accuracy. We have used $\epsilon = 0.2$ for our experiment.

A.4.2 Impact of Class Priors

Table 8: Ablation on class priors for iNaturalist, DomainNet and Tiny-ImageNet datasets. We report the Mean Accuracy (%)

Prompt	iNaturalist	DomainNet	Tiny-ImageNet
Shared + CCMP Without CP	$54.38_{\pm 0.56}$	$86.34_{\pm 0.52}$	$81.08_{\pm 0.13}$
Shared + CCMP With CP	$63.48_{\pm 1.10}$	$89.15_{\pm 0.70}$	$83.44_{\pm 0.02}$

The ablation results in the Table 8 highlight the effect of incorporating class priors into the Shared+CCMP strategy. For both iNaturalist and DomainNet, adding class priors consistently improves mean accuracy

[4] The Worst Acc for iNaturalist is the minimum accuracy of the final 250 rounds. This is reported due to the massive clients and a very low participation rate. Therefore, the worst client accuracy is zero for all methods.

compared to using Shared+CCMP without priors, with gains of nearly 9% on iNaturalist and about 3% on DomainNet.

A.4.3 Impact of CCMP and Shared Prompts

Table 9: Ablation on prompts for iNaturalist, DomainNet and Tiny-ImageNet datasets.

Prompt	iNaturalist	DomainNet	Tiny-ImageNet
Only Shared	$52.22_{\pm 0.50}$	$84.23_{\pm 0.72}$	$79.02_{\pm 0.34}$
Shared + CCMP	$63.48_{\pm 1.10}$	$89.15_{\pm 0.70}$	$83.44_{\pm 0.02}$

The ablation results in the Table 9 compare the effect of using only shared prompts versus combining them with CCMP. On iNaturalist, the mean accuracy improves from 52.22% to 63.48%, while on DomainNet, the performance rises from 84.23% to 89.15% when CCMP is added.

A.4.4 Impact of increasing shared prompts

Table 10: Impact of Accuracy on increasing the number of shared prompts with non-iid partitioning of $Dir(0.3)$. Increasing n_S results in minor improvements for CIFAR-100 and DomainNet

Dataset	$n_S = 1$	$n_S = 5$	$n_S = 10$
CIFAR-100	$88.75_{\pm 0.25}$	$89.65_{\pm 0.15}$	$90.53_{\pm 0.59}$
DomainNet	$89.15_{\pm 0.70}$	$89.29_{\pm 0.66}$	$90.22_{\pm 0.33}$

In the Table 10, we show the impact of varying the number of shared prompts. It can be observed that the impact is quite minimal.

A.4.5 On the Gain of CCMP

Table 11: Effect of increasing number of prompts in FedVPT baseline. The accuracy saturates despite increasing parameter space, indicating that gains from our method are not due to higher parameter count.

Number of Prompts	Mean Accuracy (%)
1	$83.62_{\pm 0.02}$
50	$87.15_{\pm 0.14}$
100	$87.45_{\pm 0.11}$

Introducing class-specific prompts increases the total parameter space compared to using a single global prompt. However, the performance gain achieved by our method is not solely due to this increased parameter count. To validate this, we augment the FedVPT baseline by adding 50 and 100 prompts (matching the scale of our class prompts). The mean accuracy improves initially but quickly saturates, with only marginal gains between 50 and 100 prompts. We can see this in Table 11, we have used the CIFAR-100 dataset with pathological data partitioning for this experiment. This indicates that merely increasing the number of prompt tokens is not sufficient to achieve better performance. Instead, our method's distinct soft mixing of class-specific prompts using global class prototypes and local client priors plays a key role in boosting accuracy, demonstrating the effectiveness of our proposed personalized prompt tuning mechanism.

A.4.6 Varying the Location of CCMP Injection

In the Table 12. We perform the analysis of our method PEP-FedPT. It can be seen that adding the CCMP prompts too early in the ViT is not beneficial as the `cls` token representations at the very early layers do not have better representations. Adding the prompts at later layers is also not beneficial, even tough the `cls` tokens have better representations, since the prompts inserted are not deep enough to learn useful representations. The choice of using the three prompts is to be efficient and, at the same time, to provide a fair comparison with methods like SGPT (Deng et al., 2024).

Table 12: Impact of adding the proposed CCMP prompts at different layers of ViT on CIFAR-100.

Position of CCMP	Mean Accuracy
1, 2, 3	$90.05_{\pm 0.21}$
5, 6, 7	$95.46_{\pm 0.16}$
8, 9, 10	$93.55_{\pm 0.14}$

A.4.7 Heldout Evaluation on DomainNet and iNaturalist

The comparison shows that methods like Fed-VPT-D and SGPT provide competitive results, especially on DomainNet. However, our method achieves the best overall performance, with 62.41% participating and 54.16% testing accuracy on iNaturalist, and 90.32% participating and 88.73% testing accuracy on DomainNet. This highlights its robustness across both datasets and evaluation settings. For iNaturalist about 916 clients participated in the training while 102 clients were held out. For DomainNet, 6 clients, one per domain, were held out, and 54 clients, 9 from each domain, participated in the training.

Table 13: Comparison of methods on iNaturalist and DomainNet datasets with the held-out setting

Method	iNaturalist (↑)		DomainNet (↑)	
	Participating Acc	Testing Acc	Participating Acc	Testing Acc
Head	$48.87_{\pm 0.41}$	$45.27_{\pm 0.51}$	$82.34_{\pm 1.81}$	$83.19_{\pm 2.02}$
Fed-VPT	$51.69_{\pm 0.41}$	$48.05_{\pm 0.12}$	$82.92_{\pm 1.33}$	$83.68_{\pm 1.46}$
Fed-VPT-D	$57.13_{\pm 1.12}$	$53.20_{\pm 1.18}$	$87.08_{\pm 1.24}$	$87.54_{\pm 1.52}$
P-PT	$43.87_{\pm 0.94}$	$41.20_{\pm 1.40}$	$82.89_{\pm 0.28}$	$83.05_{\pm 2.15}$
FedPR	$38.62_{\pm 0.16}$	$36.03_{\pm 0.15}$	$83.59_{\pm 0.17}$	$82.62_{\pm 1.79}$
SGPT	$55.82_{\pm 0.12}$	$53.81_{\pm 0.12}$	$86.55_{\pm 0.58}$	$87.27_{\pm 0.69}$
pFedPG	$55.61_{\pm 0.12}$	$03.05_{\pm 1.19}$	$88.34_{\pm 0.05}$	$29.40_{\pm 5.36}$
PEP-FedPT(Ours)	$\mathbf{62.41}_{\pm 0.15}$	$\mathbf{54.16}_{\pm 0.39}$	$\mathbf{90.32}_{\pm 0.18}$	$\mathbf{88.73}_{\pm 0.63}$

A.5 Visualization

A.5.1 Accuracy Vs Communication Rounds

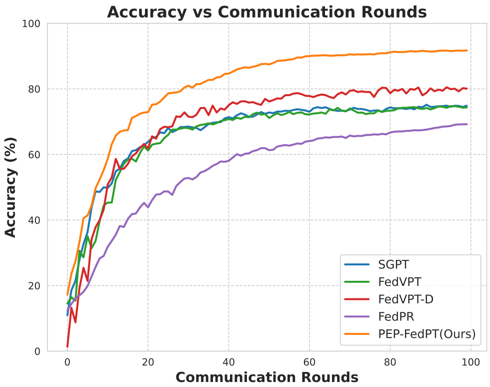


Figure 6: Comparison of the convergence of different methods across the Communication rounds on the Tiny-ImageNet dataset with pathological non-iid partitioning where each client only observes 10 classes.

The Figure 6 shows how the accuracy is improving across the FL communication rounds across the various algorithms. It is clearly evident that our proposed PEP-FedPT algorithm attains the best accuracy in fewer communication rounds compared to the other algorithms, thus minimizing the computation and communication costs.

A.5.2 t-SNE visualization of class-prompts

In the Figure 7, we show the t-sne visualization of the trained class prompts on CIFAR-100 pathological 10-class setting and we observe that each class prompt learns its own representation, which is beneficial to making the final classification decision.

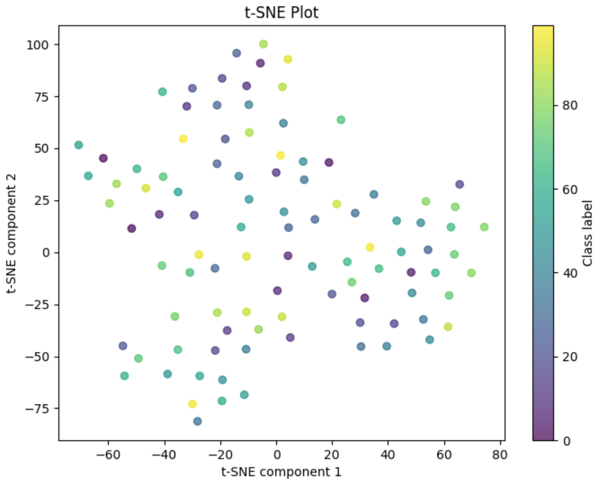
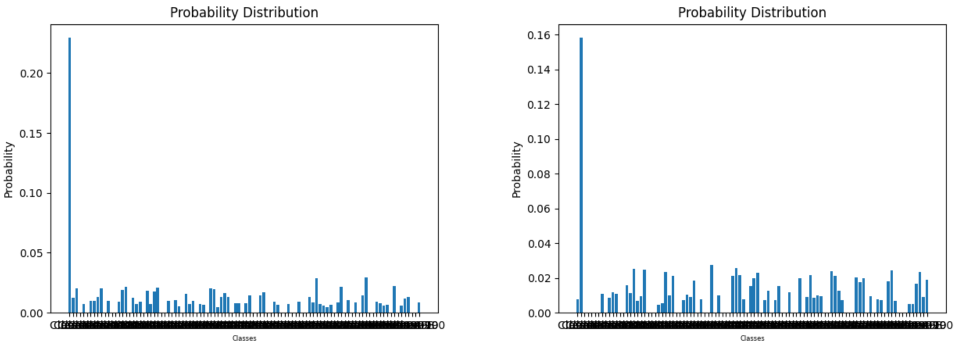


Figure 7: t-SNE visualization of the learned class prompts, it can be seen that each prompt learns its own representation implying no collapse of dimensions.

A.5.3 Visualization of the soft weights for CCMP

In the Figure 8 we plot the soft weights averaged across all the test examples belonging to class 0 and class 1 across all the clients. It can be observed that on an average the soft scores gives high score for the relevant class prompts.



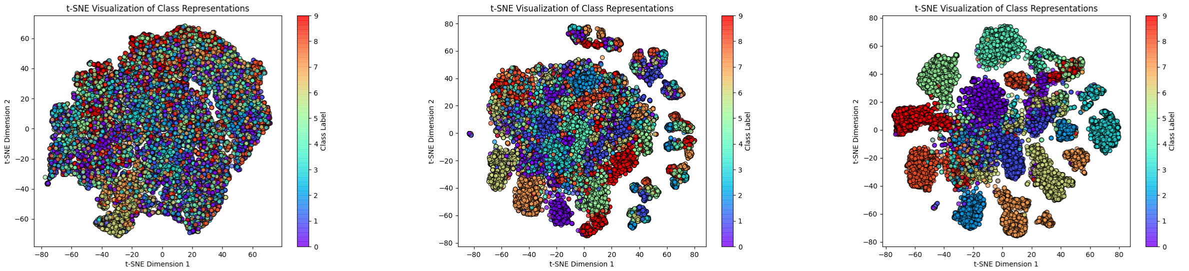
(a) class:0

(b) class:1

Figure 8: soft weights Averaged over all the data points that belong to class 0 and 1. It shows that on Average the soft weights give more importance to the prompt corresponding to the true class.

A.5.4 Visualization of Representations at different layers

In the Figures 9d and 9h, we observe that in initial layers the representations are uniformly distributed across the manifold post-training, which suggests that the utility of shared prompts is distinct from that of CCMP.

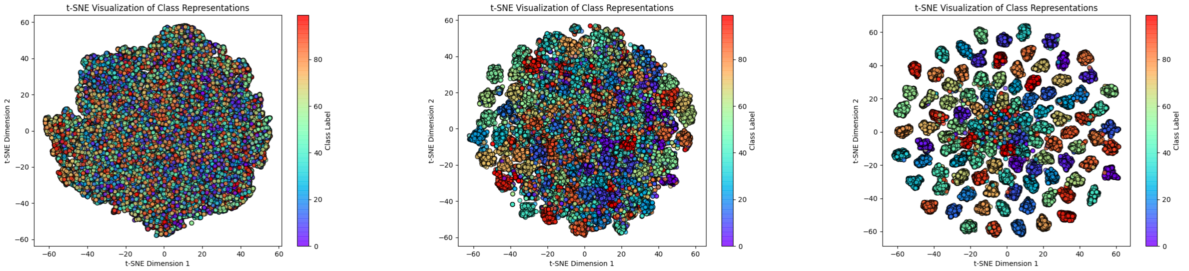


(a) Layer-0

(b) Layer-7

(c) Layer-11

(d) t-SNE representations of `cls` tokens for different layers using DomainNet dataset. It indicates that the initial layer representations are distributed uniformly over the manifold. The representation gets better once CCMP is incorporated in the later layers.



(e) Layer-0

(f) Layer-7

(g) Layer-11

(h) t-SNE representations of `cls` tokens for different layers using CIFAR-100 dataset, denoting that shared prompts at layer 0 learn only common class representations, unlike CCMP introduced later in the model. The representations are obtained using the fine-tuned ViT-B/16.

Figure 9: Comparison of t-SNE representations across layers for DomainNet and CIFAR-100 datasets.

A.6 Theoretical Details

A.6.1 CCMP as minimizer of quadratic upper bound around class prompts

For clarity and completeness, we restate the relevant proposition from the main paper.

Proposition 3. *If the assumptions 1 to 3 hold, we show that f can be upper bounded as $f \leq \tilde{L} = \frac{1}{n}\sum_{k=1,i=1}^{n,|C|} \delta_k^i \left(l_k^i(\mathbf{p}_{c_i}) + \frac{\beta_{\max}}{2} \|\mathbf{m}(k) - \mathbf{p}_{c_i}\|^2 \right) + C$ and it is minimized at $\mathbf{m}(k) = \sum_{i=1}^{|C|} \delta_k^i \mathbf{p}_{c_i}, \quad \forall k \in [n]$. which is equivalent to the (CCMP) described in sec.4.2 as $\tau >> 1$. $\beta_{\max} = \max_{i \in [|C|]} \beta_i$, C is a constant which depends on $\mathcal{P}$. This vanishes when $\mathbf{p}_{c_i} = \mathbf{p}_{c_i}^* \forall i \in [|C|]$ which makes $\tilde{L}$ a tight upper bound of f.*

Proof. We begin by applying the smoothness assumption on the loss function ℓ_k^i for each class i. By Assumption 2, ℓ_k^i is β_i-smooth, which implies that for prompts $\mathbf{m}(k) \in \mathcal{P}$ and $\mathbf{p}_{c_i} \in \mathcal{P}$, for $k \in [n]$ and $i \in [|C|]$ we have

$$\ell_k^i(\mathbf{m}(k)) \leq \ell_k^i(\mathbf{p}_{c_i}) + \nabla \ell_k^i(\mathbf{p}_{c_i})^\top (\mathbf{m}(k) - \mathbf{p}_{c_i}) + \frac{\beta_i}{2} \|\mathbf{m}(k) - \mathbf{p}_{c_i}\|^2 \tag{26}$$

$$\text{We define} \quad \beta_{\max} = \max_{i \in [|C|]} \beta_i \tag{27}$$

which gives us

$$\ell_k^i(\mathbf{m}(k)) \leq \ell_k^i(\mathbf{p}_{c_i}) + \nabla \ell_k^i(\mathbf{p}_{c_i})^\top(\mathbf{m}(k) - \mathbf{p}_{c_i})) + \frac{\beta_{\max}}{2}\|\mathbf{m}(k) - \mathbf{p}_{c_i})\|^2 \tag{28}$$

Now we know that $\mathcal{P}$ is compact, let the diameter be $D \coloneqq \sup\limits_{x,y \in \mathcal{P}} \|\mathbf{x} - \mathbf{y}\|$ which gives us

$$\|\mathbf{p}_1 - \mathbf{p}_2\| \leq D \quad \text{for all } \mathbf{p}_1, \mathbf{p}_2 \in \mathcal{P} \tag{29}$$

$$\Rightarrow \|\mathbf{m}(k) - \mathbf{p}_{c_i}\| \leq D \tag{30}$$

Since by Assumption 1 $l^i(\mathbf{p})$ is β_i-smooth, we have for $\mathbf{x}, \mathbf{y} \in \mathcal{P}$

$$\|\nabla \ell_k^i(\mathbf{x}) - \nabla \ell_k^i(\mathbf{y})\| \leq \beta_i \|\mathbf{x} - \mathbf{y}\| \tag{31}$$

$$\leq \beta_{\max}\|\mathbf{x} - \mathbf{y}\| \quad \text{From equation 27} \tag{32}$$

$$\text{If} \quad \forall \delta \geq 0 \quad \|\mathbf{x} - \mathbf{y}\| \leq \delta, \quad \text{for} \quad \epsilon = \delta\beta_{\max} \quad \text{we have}$$

$$\|\nabla \ell_k^i(\mathbf{x}) - \nabla \ell_k^i(\mathbf{y})\| \leq \epsilon \Rightarrow \|\nabla \ell_k^i(\mathbf{x})_j - \nabla \ell_k^i(\mathbf{y})_j\| \leq \epsilon, j \in [d] \tag{33}$$

$\nabla \ell_k^i(\mathbf{x})_j$ is a continuous mapping of compact metric space $\mathcal{P}$ into metric space $\mathbb{R}$

$$\Rightarrow \nabla \ell_k^i(\mathbf{x})_j \quad \text{is compact} \quad \forall j \in [d]$$

Let B_{i_j} be the diameter of $\nabla \ell_k^i(\mathcal{P})_j$, $j \in [d]$, then

$$\|\nabla \ell_k^i(\mathbf{p}_{c_i})\| \leq B_i = \sum_{j=1}^{d} |B_{i_j}| \tag{34}$$

Using Cauchy-Schwartz inequality and from 34 & 30 we have

$$\nabla \ell_k^i(\mathbf{p}_{c_i})^\top(\mathbf{m}(k) - \mathbf{p}_{c_i}) \leq DB_i \tag{35}$$

From 28

$$\ell_k^i(\mathbf{m}(k)) \leq \ell_k^i(\mathbf{p}_{c_i}) + \tilde{C}_k + \frac{\beta_{\max}}{2}\|\mathbf{m}(k) - \mathbf{p}_{c_i}\|^2 \tag{36}$$

where $\tilde{C}_k = DB_i$. The global loss of the clients is given by

$$L = \frac{1}{n}\sum_{k=1}^{n}\left(\sum_{i=1}^{|C|} \delta_k^i \cdot \ell_k^i(\mathbf{m}(k))\right) \tag{37}$$

$$\leq \tilde{L} = \frac{1}{n}\sum_{k=1}^{n}\left[\sum_{i=1}^{|C|} \delta_k^i \left(\ell_k^i(\mathbf{p}_{c_i}) + \frac{\beta_{\max}}{2}\left\|\mathbf{m}(k) - \mathbf{p}_{c_i}\right\|^2\right)\right] + \tilde{C} \tag{38}$$

$$\text{where} \quad \tilde{C} = \frac{1}{n}\sum_{k=1}^{n}\tilde{C}_k \tag{39}$$

which proves the first part of our main proposition 4 in the paper.
If $\mathbf{p}_{c_i} = \mathbf{p}_{c_i}^*$, we have a tight upper bound $\tilde{L} = \frac{1}{n}\sum_{k=1}^{n}\left[\sum_{i=1}^{|C|} \delta_k^i \left(\ell_k^i(\mathbf{p}_{c_i}) + \frac{\beta_{\max}}{2}\left\|\mathbf{m}(k) - \mathbf{p}_{c_i}\right\|^2\right)\right]$, because

$\nabla \ell_k^i(\mathbf{p}_{c_i})$ vanishes, according to Assumption 3.
We are interested in finding the optimal client prompts $\mathbf{m}(k)$ for each client k.

$$\frac{\partial \tilde{L}_k}{\partial \mathbf{m}(k)} = \frac{1}{n} \sum_{i=1}^{|C|} \delta_k^i \beta_{\max} \left(\mathbf{m}(k) - \mathbf{p}_{c_i}\right) \tag{40}$$

$$= \frac{\beta_{\max}}{N} \left(\mathbf{m}(k) - \sum_{i=1}^{|C|} \delta_k^i \mathbf{p}_{c_i}\right), \quad \text{since} \sum_{i=1}^{|C|} \delta_k^i = 1 \tag{41}$$

$$\text{Setting } \frac{\partial \tilde{L}_k}{\partial \mathbf{m}(k)} = 0, \text{ we have} \quad \mathbf{m}(k) = \sum_{i=1}^{|C|} \delta_k^i \mathbf{p}_{c_i} \tag{42}$$

which gives the second part of our proposition 4, and completes our proof. □

A.6.2 CCMP as MMSE estimator of the true class prompt

Proposition 2. *If the cls tokens and the class-specific prompts at input of layer l has the joint density given by $p_k(\mathbf{cls}_{l-1}, \mathbf{p})$ as in Eq. 17, then the CCMP prompt for a client k, $\mathbf{m}_{l-1}(k)$ obtained in Eq. 6 is Minimum Mean Squared Estimator (MMSE) of the true class prompt.*

Proof. Consider the following mean-squared error

$$J(\hat{\mathbf{p}}) = \mathbb{E}\|\mathbf{p} - \hat{\mathbf{p}}\|^2 \tag{43}$$

where the expectation is taken across the joint distribution of $p_k(\mathbf{p}, \mathbf{cls}_{l-1})$. The $\hat{\mathbf{p}}$ that's minimizes the $J(\hat{\mathbf{p}})$ is the MMSE estimator, and $\mathbf{p}$ is our true class prompt. We have the following

$$\begin{aligned} J(\hat{\mathbf{p}}) &= \mathbb{E}\|\mathbf{p} - \hat{\mathbf{p}}\|^2 \\ &= \mathbb{E}\|\mathbf{p} - \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}] + \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}] - \hat{\mathbf{p}}\|^2 \\ &= \mathbb{E}\|\mathbf{p} - \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}]\|^2 + \mathbb{E}\|\mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}] - \hat{\mathbf{p}}\|^2 \\ &+ 2\mathbb{E}\langle \mathbf{p} - \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}], \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}] - \hat{\mathbf{p}}\rangle \\ &= \mathbb{E}\|\mathbf{p} - \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}]\|^2 + \mathbb{E}\|\mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}] - \hat{\mathbf{p}}\|^2 \end{aligned}$$

The equality is obtained as the cross term is zero i.e we have $\mathbb{E}[\langle \mathbf{p} - \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}], \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}] - \hat{\mathbf{p}}\rangle] = 0$. It follows by using the iterated expectation as shown below.

$$\begin{aligned} \mathbb{E}\langle \mathbf{p} - \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}], \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}] - \hat{\mathbf{p}}\rangle &= \mathbb{E}[\mathbb{E}[\langle \mathbf{p} - \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}], \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}] - \hat{\mathbf{p}}\rangle|\mathbf{cls}_{l-1}]] & (44) \\ &= \mathbb{E}[\mathbb{E}[\langle \mathbf{p} - \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}]|\mathbf{cls}_{l-1}, \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}] - \hat{\mathbf{p}}\rangle]] & (45) \\ &= 0 & (46) \end{aligned}$$

We now have

$$J(\hat{\mathbf{p}}) = \mathbb{E}\|\mathbf{p} - \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}]\|^2 + \mathbb{E}\|\mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}] - \hat{\mathbf{p}}\|^2 \tag{47}$$

From the above Eq. 47 it can be readily seen that $J(\hat{\mathbf{p}})$ is minimized by setting the value of $\hat{\mathbf{p}} = \mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}]$

$$\mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}] = \sum_{m=1}^{|C|} p(\mathbf{p} = \mathbf{p}_{c_m}|\mathbf{cls}_{l-1}) \cdot \mathbf{p}_{c_m} \tag{48}$$

From Eq. 16, we can rewrite the above equation

$$\begin{aligned}\mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}] &= \sum_{m=1}^{|C|} s^m_{i,l-1,k} \cdot \mathbf{p}_{c_m} \\ &= \mathbf{P}_C * \mathbf{s}_{i,l-1,k}\end{aligned}$$

From Eq. 25 we conclude that $\mathbb{E}[\mathbf{p}|\mathbf{cls}_{l-1}] = \mathbf{m}_{l-1}$ □

A.6.3 Convergence

We assume the following assumptions on the loss functions based on (Karimireddy et al., 2020; Acar et al.).

A 4. *The loss functions f_k are Lipschiltz smooth, i.e., $\|\nabla f_k(\boldsymbol{\theta}_1) - \nabla f_k(\boldsymbol{\theta}_2)\| \leq \beta \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|$.*

A 5. $\frac{1}{n}\sum_{k\in[n]} \|\nabla f_k(\boldsymbol{\theta})\|^2 \leq G^2 + B^2\|\nabla f(\boldsymbol{\theta})\|^2$,*where* $f(\boldsymbol{\theta}) = \frac{1}{n}\sum_{k\in[n]} f_k(\boldsymbol{\theta})$.*This is referred to bounded gradient dissimilarity assumption,*

A 6. *let $\mathbb{E}\|\nabla l(\boldsymbol{\theta}, (x, y)) - \nabla f_k(\boldsymbol{\theta})\| \leq \sigma^2$, for all k and $\boldsymbol{\theta}$. Here $l(\boldsymbol{\theta}, (x, y))$ is loss evaluated on the sample (x, y) and $f_k(\boldsymbol{\theta})$ is expectation across the samples drawn from $\mathcal{D}_k$. This is a bounded variance assumption.*

In the above assumptions, the parameter $\boldsymbol{\theta}$ denotes the trainable, shared, and class-specific prompts along with the classification head parameters.

The entire computation of the soft scores $\mathbf{s}_{i,l-1,k}$ for the client k, based on `cls`, can be viewed as a part of the model architecture itself (Fig.1) and encapsulated inside the client's loss function.

We then have the following proposition.

Proposition 3. *Theorem V of Karimireddy et al. (2020) in Appendix D.2: let $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$, the global step-size be α_g and the local step-size be α_l. When the update period R is very large or $\tau >> 1$, the PEP-FedPT algorithm will have contracting gradients. If Initial model is $\boldsymbol{\theta}^0$, $F = f(\boldsymbol{\theta}^0) - f(\boldsymbol{\theta}^*)$ and for constant M, then in T rounds, the model $\boldsymbol{\theta}^T$ satisfies $\mathbb{E}[\|\nabla f(\boldsymbol{\theta}^T)\|^2] \leq O(\frac{\beta M\sqrt{F}}{\sqrt{RLS}} + \frac{\beta^{1/3}(FG)^{2/3}}{(T+1)^{2/3}} + \frac{\beta B^2 F}{T})$.*

The above proposition states that the PEP-FedPT algorithm requires $\mathcal{O}(\frac{1}{\epsilon^2})$ communication rounds to make the average gradients of the global model smaller, i.e., $\mathbb{E}[\|\nabla f(\boldsymbol{\theta}^T)\|^2] \leq \epsilon$. The result is plug and play because we only employ global prompts and parameters for the training.

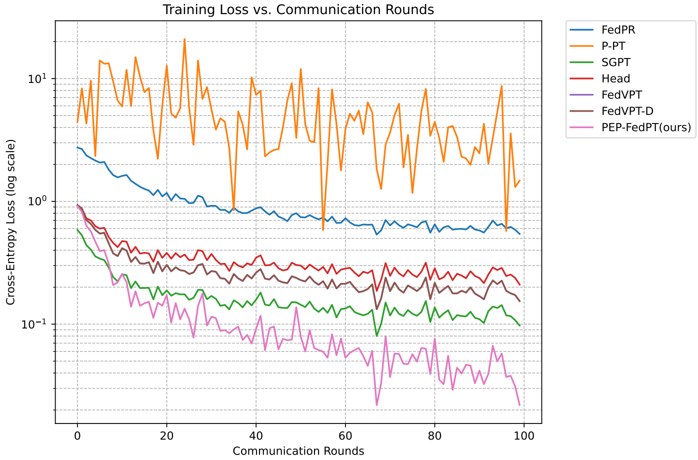


Figure 10: Comparison of training loss of various algorithms on CIFAR-100 dataset

Figure 10 illustrates the training loss (cross-entropy, log scale) versus communication rounds for various baselines. We observe that P-PT struggles to converge, while FedPR and Head show limited improvements with early plateauing. Methods such as SGPT and FedVPT achieve more stable convergence, and FedVPT-D further reduces the loss by incorporating additional regularization. In contrast, our proposed PEPFedPT consistently outperforms all baselines, achieving both faster convergence and the lowest final loss. Our theory, which minimizes the quadratic upper bound at convergence is expoected to have lower training loss. This empirical trend aligns with our predictions, thereby ensuring improved stability and convergence in practice.

A.6.4 Analysis of CCMP when the scores are the function of data

We show that CCMP minimizes the quadratic upper bound on the loss even when the scores are functions of both the data and class-priors. We denote the estimate of class prompt for class i at any round to be $\mathbf{p}_{c_i}$. We denote the class prompts by $\mathbf{P}_C = [\mathbf{p}_{c_1}, \mathbf{p}_{c_2} \dots, \mathbf{p}_{c_{|C|}}]$. Let $\mathbf{m}(k, \mathbf{x})$ denote the prompt used at client k for data point $\mathbf{x}$. [[5]], and let the total number of clients be n, and δ_k^i denote the empirical probability that a data point at client k belongs to class i. We assume that the joint density of the data in client $p_k(\mathbf{x}, y)$ is modeled as $p_k(\mathbf{x}, y) := p_k(\mathbf{x})p_k(y|\mathbf{x})$, the posterior $p_k(y|\mathbf{x})$ is assumed to be given by the scores in Eq. 15 which we denote by $s_{k,i,\mathbf{x}}$ and we model $p_k(y = i|\mathbf{x})$ by defining $p_k(y = i|\mathbf{x}) := s_{k,i,\mathbf{x}}$. Let $\mathcal{P}$ be the set of all possible prompts across all the clients, such that $\mathbf{m}(k, \mathbf{x}) \in \mathcal{P}, \quad \forall k \in \{1, 2, \dots, n\} \quad , \forall \mathbf{x}$. The overall loss of the client k is denoted by the $\mathbb{E}[l_k(\mathbf{m}(k, \mathbf{x}), \mathbf{x}, y)]$. Note the expectation is over the $p_k(\mathbf{x}, y)$. The goal is to estimate $\mathbf{m}(k, \mathbf{x})$ as a function of class prompts $\{\mathbf{p}_{c_1}, \mathbf{p}_{c_2} \dots, \mathbf{p}_{c_{|C|}}\}$. The global loss across all clients can be computed as $f = \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}[l_k(\mathbf{m}(k, \mathbf{x}), \mathbf{x}, y)]$.

We now state the following assumptions:

A 7. *$\mathcal{P}$ is compact subset of $\mathbb{R}^d$, where d is the token dimension.*

A 8. *$l_k(\boldsymbol{\theta}, \mathbf{x}, y)$ is β smooth in argument $\boldsymbol{\theta}$ with parameter β $\forall y \in [|C|]$,$\forall \mathbf{x}$, $\forall k \in [n]$.*

Proposition 4. *If $\ell_k(\boldsymbol{\theta}, (\mathbf{x}, y))$ satisfies the above assumptions 7 to 8, we show that overall loss function $f = \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}[\ell_k(\mathbf{m}(k, \mathbf{x}), (\mathbf{x}, y))]$ can be upper bounded as $f \leq \tilde{L} = \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\Big[\sum_{i=1}^{|C|} s_k^i \left(\ell_k^i(\mathbf{p}_{c_i}) + \frac{\beta_{\max}}{2}\left\|\mathbf{m}(k) - \mathbf{p}_{c_i}\right\|^2\right)\Big] + \tilde{C}$ and it is minimized at $\mathbf{m}(k) = \sum_{c=1}^{|C|} s_k^i \mathbf{p}_{c_i}, \quad \forall k \in [n]$. which is equivalent to the (CCMP) described in sec.4.2 . $\tilde{C}$ is a constant which depends on $\mathcal{P}$. The $\mathbb{E}$ is over the distribution of the data $\mathbf{x}$. Here we defined $\mathbf{m}(k) := \mathbf{m}(k, \mathbf{x})$, $\ell_k^i(\boldsymbol{\theta}) := \ell_k(\boldsymbol{\theta}, \mathbf{x}, y = i)$ and $s_k^i := s_{k,i,\mathbf{x}}$*

Proof. we expand the clients loss $\mathbb{E}[\ell_k(\mathbf{m}(k, \mathbf{x}), \mathbf{x}, y)]$ as below

$$\begin{aligned}
\mathbb{E}[\ell_k(\mathbf{m}(k, \mathbf{x}), \mathbf{x}, y)] &= \mathbb{E}[\mathbb{E}[\ell_k(\mathbf{m}(k, \mathbf{x}), \mathbf{x}, y)]|\mathbf{x}] && (49)\\
&= \mathbb{E}[\sum_{i=1}^{|C|} \ell_k(\mathbf{m}(k, \mathbf{x}), \mathbf{x}, y = i)p_k(y = i|\mathbf{x})] && (50)\\
&= \mathbb{E}[\sum_{i=1}^{|C|} \ell_k(\mathbf{m}(k, \mathbf{x}), \mathbf{x}, y = i)s_{k,i,\mathbf{x}}] && (51)\\
&= \mathbb{E}[\sum_{i=1}^{|C|} \ell_k^i(\mathbf{m}(k))s_k^i] && (52)
\end{aligned}$$

In the last step we use the definitions in the proposition i.e, $\mathbf{m}(k) := \mathbf{m}(k, \mathbf{x})$, $\ell_k^i(\mathbf{m}(k)) := \ell_k(\mathbf{m}(k, \mathbf{x}), \mathbf{x}, y = i)$ and $s_k^i := s_{k,i,\mathbf{x}}$.

If the Lipschitz smooth(8) and compactness(7) assumptions hold, then by following similar arguments from 26 till 38 we will have the global loss of clients given by,

[5] for notation convenience, we drop the layer index j from $\mathbf{m}_j(k, x)$.

$$f = \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\sum_{i=1}^{|C|} s_k^i \cdot \ell_k^i(\mathbf{m}(k))\right] \tag{53}$$

$$\leq \tilde{L} = \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\sum_{i=1}^{|C|} s_k^i \left(\ell_k^i(\mathbf{p}_{c_i}) + \frac{\beta}{2}\left\|\mathbf{m}(k) - \mathbf{p}_{c_i}\right\|^2\right)\right] + \tilde{C} \tag{54}$$

which proves the first part.

We are interested in finding the optimal client prompts $\mathbf{m}(k)$ for each client k and for each data point $\mathbf{x}$. This is obtained by optimizing the argument inside the expectation which is $\left[\sum_{i=1}^{|C|} s_k^i \left(\ell_k^i(\mathbf{p}_{c_i}) + \frac{\beta}{2}\left\|\mathbf{m}(k) - \mathbf{p}_{c_i}\right\|^2\right)\right]$ with respect to $\mathbf{m}(k)$.

$$\frac{\partial \sum_{i=1}^{|C|} s_k^i \left(\ell_k^i(\mathbf{p}_{c_i}) + \frac{\beta}{2}\left\|\mathbf{m}(k) - \mathbf{p}_{c_i}\right\|^2\right)}{\partial \mathbf{m}(k)} = \frac{1}{n}\sum_{i=1}^{|C|} s_k^i \beta\left(\mathbf{m}(k) - \mathbf{p}_{c_i}\right) \tag{55}$$

$$= \frac{\beta}{N}\left(\mathbf{m}(k) - \sum_{i=1}^{|C|} s_k^i \mathbf{p}_{c_i}\right), \quad \text{since} \sum_{i=1}^{|C|} s_k^i = 1 \tag{56}$$

$$\text{Setting } \frac{\partial \tilde{L}_k}{\partial \mathbf{m}(k)} = 0, \text{ we have} \quad \mathbf{m}(k) = \sum_{i=1}^{|C|} s_k^i \mathbf{p}_{c_i} \tag{57}$$

which gives the second part of our proposition 4, and completes our proof. □