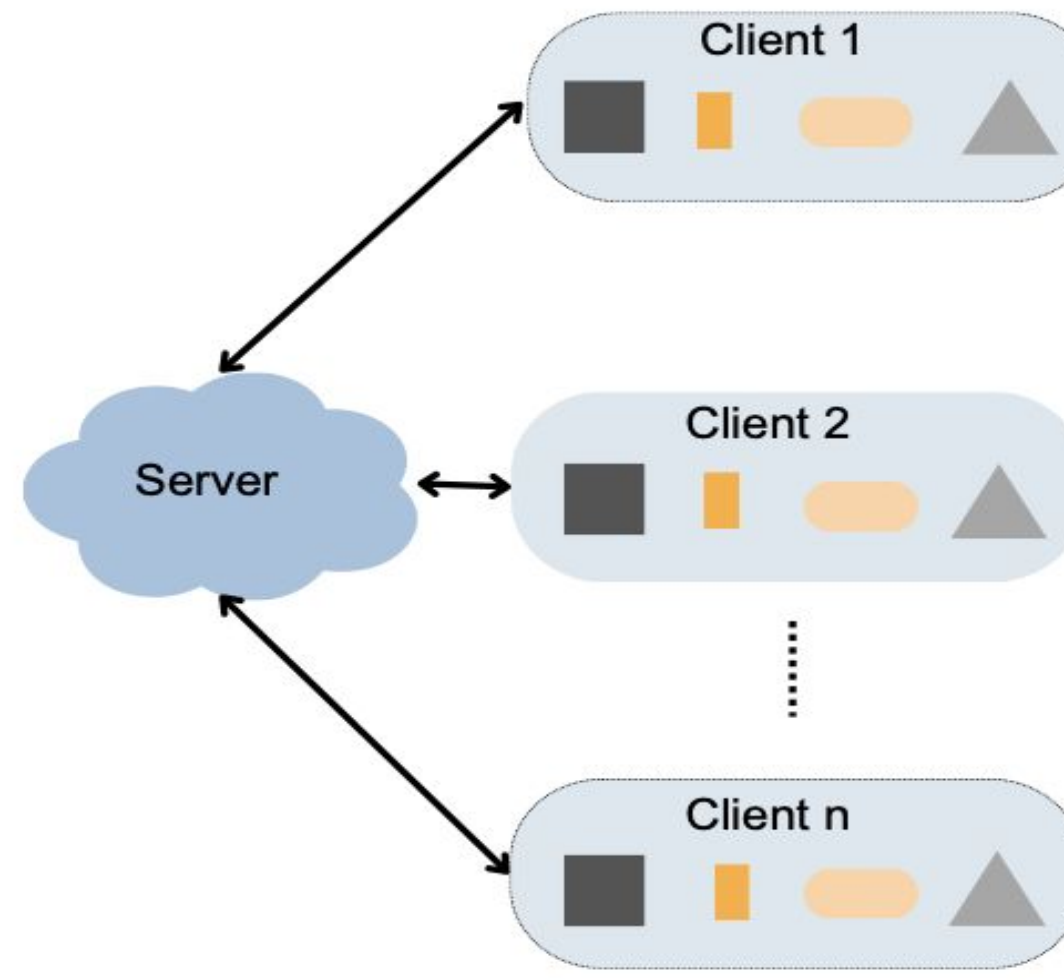




Setup and Framework for PEP-FedPT

Prompt Tuning in FL



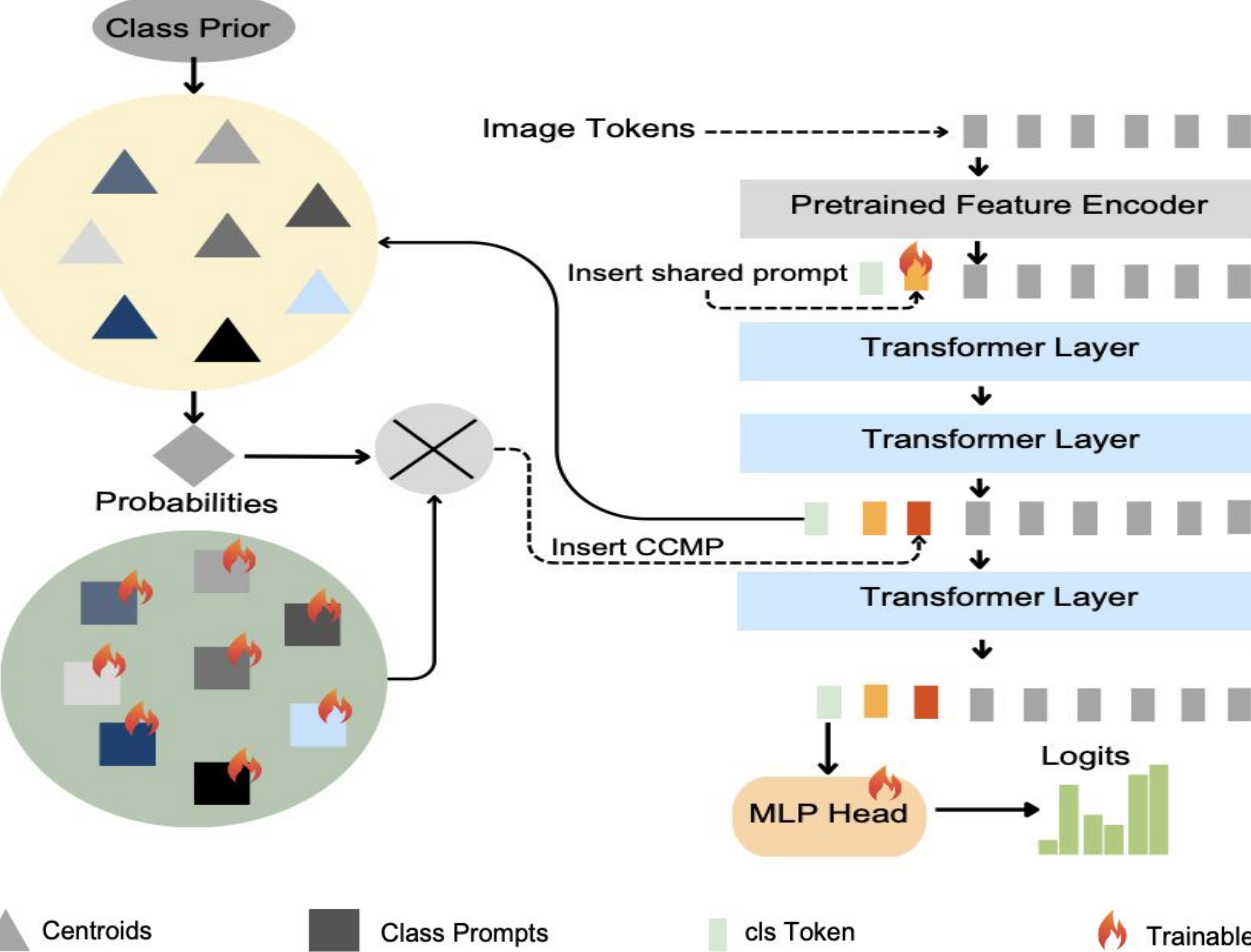
Core Challenges

- 1 **Global Prompt Tuning** - poor generalization across heterogeneous clients
- 2 **Personalized Prompt Tuning** - overfits local data, lacks global generalization

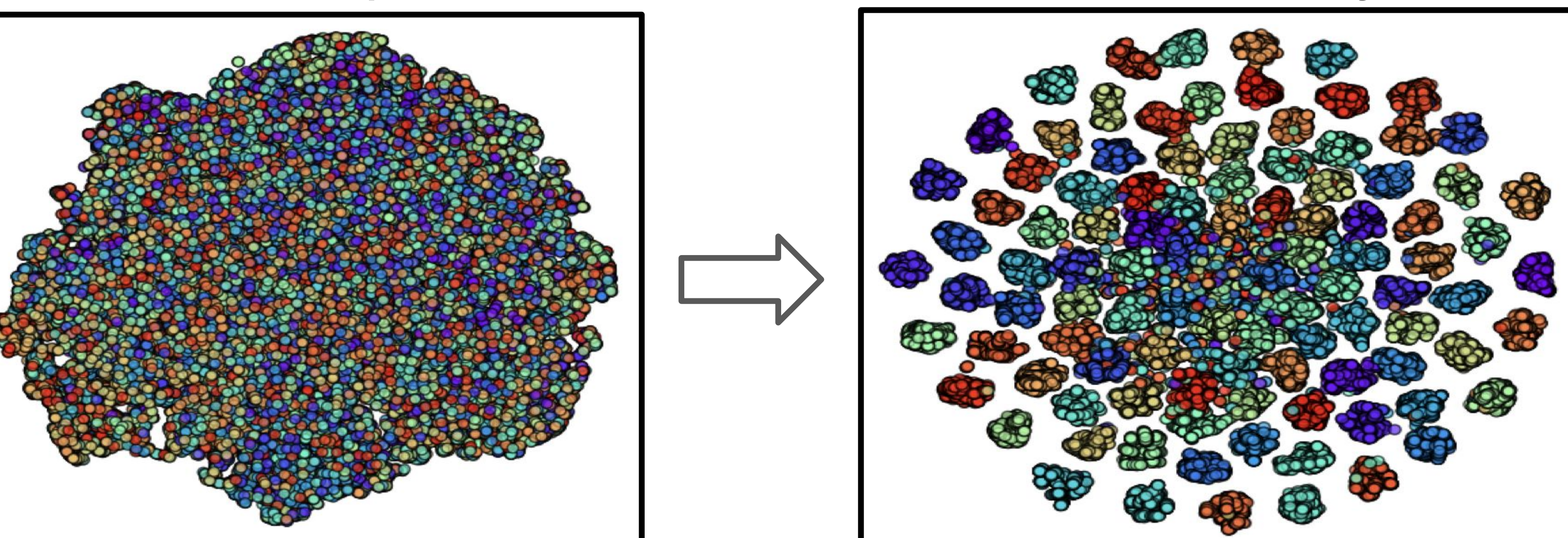
Our Solution

Class-Contextualized Mixed Prompt (CCMP) - adaptively combined from global class prototypes and client class priors & cls-token representations.

PEP-FedPT achieves generalization and sample level personalization by use of global prompts ONLY.



Cls-token representation before and after CCMP insertion at ViT layer



Method and Algorithm

Algorithm 1: Pseudocode for PEP-FedPT

- 1 Compute global class prototypes before training;
- 2 **for each communication round do**
- 3 Server sends model parameters to clients;
- 4 **for each client in parallel do**
- 5 **for local epochs do**
- 6 Insert shared prompts at layer 0;
- 7 **for layers in which CCMP is inserted do**
- 8 Compute class scores from priors and CLS-prototype similarity;
- 9 Mix scores with class prompts to form CCMP;
- 10 Insert CCMP;
- 11 Train shared and class prompts;
- 12 Send prompt updates and class prototypes to server;
- 13 Server aggregates prompt updates;
- 14 Periodically, server aggregates class prototypes;

① **Prompt insertion into a ViT layer:** The input and output can be written as

$$\mathbf{cls}_l, \mathbf{P}_{S_l}, \mathbf{E}_l = TL_l(\mathbf{cls}_{l-1}, \mathbf{P}_{S_{l-1}}, \mathbf{E}_{l-1})$$

② **Computation of class prototypes:** Using cls-tokens for a particular class

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

③ **Computation of CCMP scores:** Using client class priors and its prototypes

$$\hat{s}_{i,l-1,k}^c = \exp\left(\frac{\text{sim}(\mathbf{cls}_{l-1,i,k}, \mu_{l-1}^c)}{\tau}\right) \delta_k^c \quad \left| \quad s_{i,l-1,k}^c = \frac{\hat{s}_{i,l-1,k}^c}{\sum_{m=1}^{|C|} \hat{s}_{i,l-1,k}^m}$$

④ **Computation of CCMP:** Using CCMP scores and class specific prompts

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

⑤ **Aggregation of class prototypes:** Server aggregates periodically

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

i: data-point index, l: layer index, k: client index, r: aggregation period index, n: number of data points, c: class index

Results and Analysis

95.46%
CIFAR-100
+2.5% over
pFedPG

91.52%
Tiny-Imagenet
+8.59 over
pFedPG

89.15%
DomainNet
+1.75 over
pFedPG

63.48%
i-Naturalist
+5.52 over
Fed-VPT-D

Method	CIFAR-100	Tiny-Imagenet	DomainNet	i-Naturalist
Head-Tuning	77.85	68.39	83.71	49.41
Fed-VPT	83.62	74.20	84.23	52.22
Fed-VPT-D	85.15	79.60	87.31	57.96
FedPR	81.77	68.86	82.95	41.25
SGPT	84.16	75.65	85.56	55.78
pFedPG	92.96	82.93	87.40	52.42
P-PT	75.37	61.68	82.30	45.69
PEP-FedPT	95.46	91.52	89.15	63.48

Computation and Communication Costs

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	>8.4 M	>100
PEP-FedPT (Ours)	1153	4.6 M	12

Theoretical Underpinnings of CCMP

• Assuming prompt space is compact and the loss function is Lipschitz-smooth, **CCMP minimizes a quadratic upper bound on the loss**. The loss is minimized at $\mathbf{m}^{(k)} = \sum_c \delta_k^c \mathbf{p}_c$, and the bound becomes tight when $\mathbf{p}_c = \mathbf{p}^*$.

• CCMP ($\mathbf{m}^{(l-1)}$) is proven to be the **Minimum Mean Squared Error (MMSE) estimator** of the true class prompt, derived from a joint density $p_k(\mathbf{cls}_{l-1}, \mathbf{p})$ over cls tokens and prompts with posterior probabilities modeled from ViT outputs at layer l-1.

