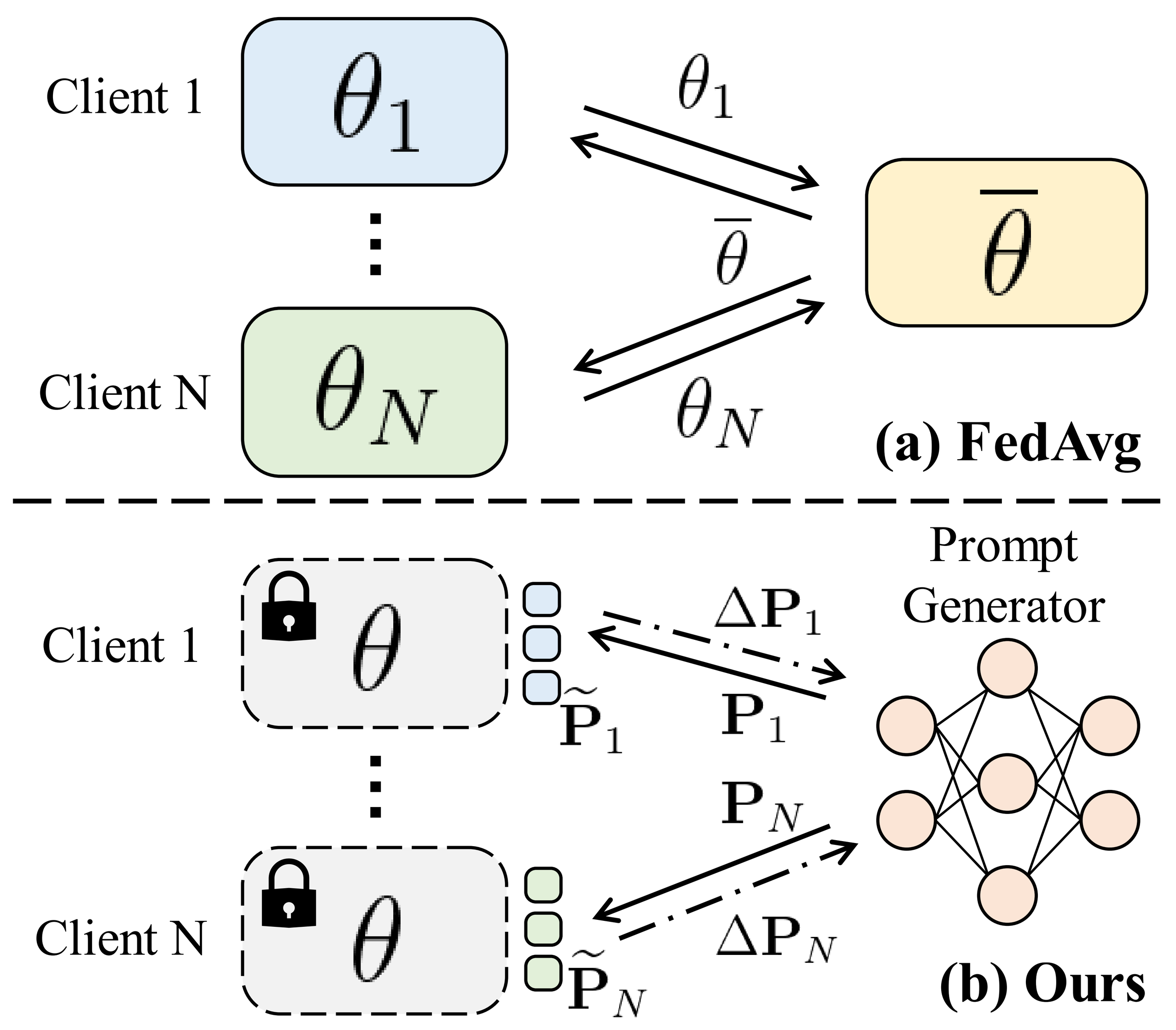
# Efficient Model Personalization in Federated Learning via Client-Specific Prompt Generation

National Taiwan University
NVIDIA

Fu-En Yang[1,2] Chien-Yi Wang[2] Yu-Chiang Frank Wang[1,2]
[1]National Taiwan University [2]NVIDIA

ICCV23
PARIS

## Introduction & Overview




### Limited computational resources & communication bandwidth

– FedAvg updates/transports entire model parameters -> high comp./comm. Cost

– Hard to incorporate large-scale backbones (ViT)

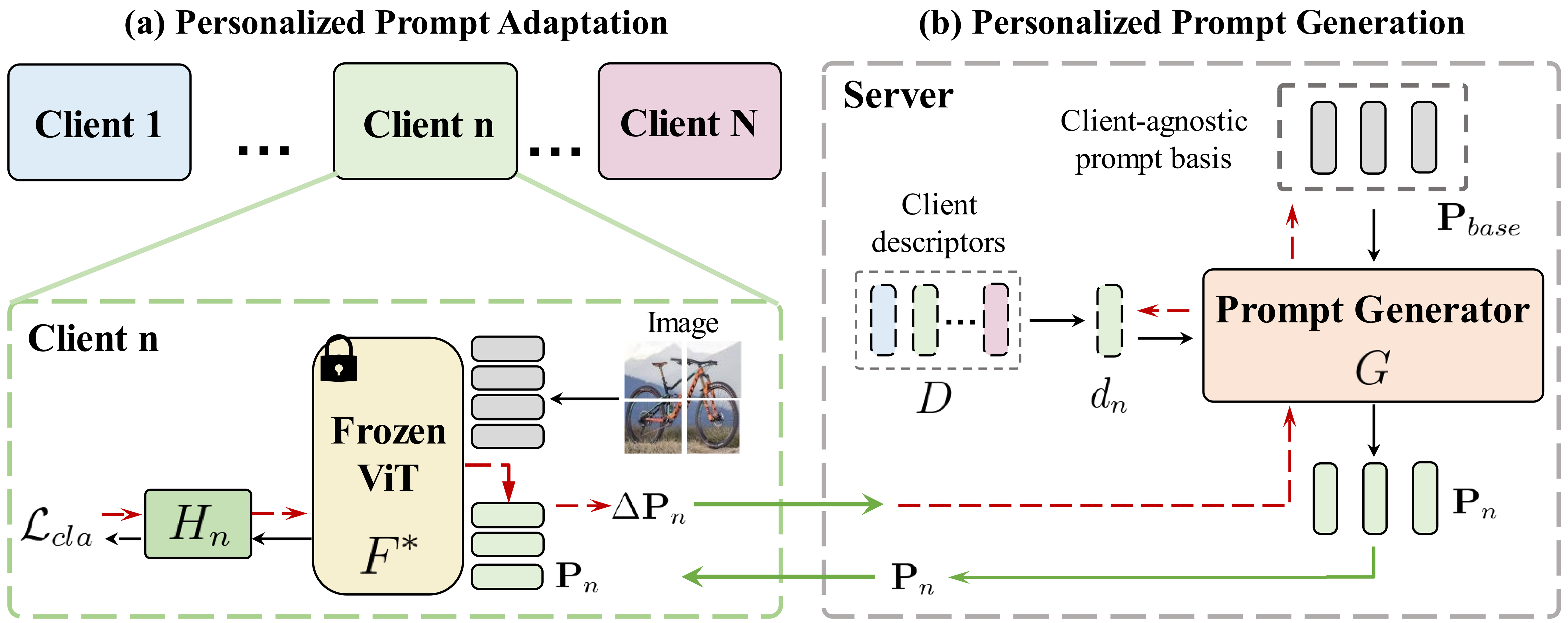
+ We adopt prompt learning -> PEFT

### Data heterogeneity exists among local clients (hospitals/edge devices)

– FedAvg simply averages models from clients that would not address data heterogeneity well

+ We learn to generate client-specific prompts for all local clients at the server

## Personalized Federated Learning via Client-Specific Prompt Generation (pFedPG)




### Personalized Prompt Adaptation @clients

- Leverage pre-trained Vision Transformer (ViT) as the backbone
- Keep backbone frozen, only update prompts ($\mathbf{P}_n$) & classifier ($H_n$)

$$\mathcal{L}_n = \frac{1}{|\mathcal{D}_n|}\sum_{j=1}^{|\mathcal{D}_n|}\mathcal{L}_{cla}\left(H_n\left(F^*\left([c, \mathbf{P}_n, \mathbf{z}_j]\right)\right), y_j\right)$$

### Personalized Prompt Generation @server

- Prompt Generator $G$ generates personalized prompts for clients
- All $G$, $\mathbf{P}_{base}$ , and $d\_n$ are updated @server by the clients' optimization directions ($\Delta\mathbf{P}_n$):

$$\Delta\varphi = \nabla_\varphi \mathcal{L}_n = (\nabla_\varphi \mathbf{P}_n)^T \nabla_{\mathbf{P}_n}\mathcal{L}_n \cong (\nabla_\varphi \mathbf{P}_n)^T \Delta\mathbf{P}_n,$$

$$\begin{aligned} \varphi &\leftarrow \varphi - \alpha \nabla_\varphi \mathbf{P}_n^T \Delta \mathbf{P}_n, \\ \mathbf{P}_{base} &\leftarrow \mathbf{P}_{base} - \alpha \nabla_{\mathbf{P}_{base}} \varphi^T \nabla_\varphi \mathbf{P}_n^T \Delta \mathbf{P}_n, \\ d_n &\leftarrow d_n - \alpha \nabla_{d_n} \varphi^T \nabla_\varphi \mathbf{P}_n^T \Delta \mathbf{P}_n. \end{aligned}$$

## Quantitative Comparisons (Domain Shift & Imbalanced Class Distribution)

| Datasets | Office-Caltech10 (%) | | | | | DomainNet (%) | | | | | | | Comm. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | *A* | *C* | *D* | *W* | Avg. | *C* | *I* | *P* | *Q* | *R* | *S* | Avg. | Cost |
| ***Baselines*** | | | | | | | | | | | | | |
| SingleSet-Full | 80.73 | 73.33 | 90.62 | 94.92 | 84.90 | 47.34 | 37.14 | 67.21 | 55.30 | 84.88 | 45.13 | 56.17 | - |
| SingleSet-VPT [21] | 83.33 | 74.67 | 96.88 | 96.61 | 87.87 | 57.98 | 41.55 | 74.64 | 59.60 | 89.56 | 60.47 | 63.97 | - |
| FedAvg [40] | 89.58 | 80.44 | 100.0 | 100.0 | 92.51 | 63.50 | 38.05 | 71.89 | 60.80 | 78.55 | 60.47 | 62.21 | $8.58 \times 10^7$ |
| ***Personalized Federated Learning*** | | | | | | | | | | | | | |
| Per-FedAvg [14] | 91.67 | 90.22 | 100.0 | 100.0 | 95.47 | 69.39 | 48.71 | 82.07 | 35.30 | 90.63 | **72.56** | 66.44 | $8.58 \times 10^7$ |
| FedRep [8] | 91.15 | 88.44 | 100.0 | 100.0 | 94.90 | 64.26 | 38.20 | 72.86 | **62.10** | 82.66 | 60.11 | 63.37 | $8.58 \times 10^7$ |
| FedRoD [4] | 92.19 | 90.67 | 100.0 | 100.0 | 95.72 | 66.54 | 42.92 | 74.15 | 57.20 | 84.63 | 66.43 | 65.31 | $8.58 \times 10^7$ |
| FedBABU [37] | 89.06 | 85.78 | 100.0 | 100.0 | 93.71 | 63.31 | 43.07 | 74.80 | 43.80 | 87.26 | 67.15 | 63.23 | $8.58 \times 10^7$ |
| ***Efficient Federated Learning*** | | | | | | | | | | | | | |
| FedVPT [21] | 92.71 | 84.44 | 100.0 | 100.0 | 94.29 | 65.59 | 44.14 | 76.58 | 47.30 | 91.04 | 60.29 | 64.16 | $7.68 \times 10^3$ |
| FedVPT-D [21] | 91.67 | 89.33 | 100.0 | 100.0 | 95.25 | 63.31 | 43.07 | 74.80 | 54.80 | 87.26 | 67.15 | 65.07 | $9.22 \times 10^3$ |
| pFedPG (Ours) | **94.79** | **92.44** | **100.0** | **100.0** | **96.81** | **73.00** | **50.08** | **84.33** | 60.00 | **94.00** | 68.41 | **71.64** | $7.68 \times 10^3$ |

| Datasets | CIFAR-10 (%) | | CIFAR-100 (%) | |
|---|---|---|---|---|
| Method | Disjoint | $Dir(0.1)$ | Disjoint | $Dir(0.1)$ |
| ***Baselines*** | | | | |
| SingleSet-Full | 89.51 | 83.85 | 67.74 | 49.64 |
| SingleSet-VPT [21] | 88.91 | 84.32 | 63.42 | 46.46 |
| FedAvg [40] | 88.04 | 79.79 | 63.33 | 51.37 |
| ***Personalized Federated Learning*** | | | | |
| Per-FedAvg [14] | 88.13 | 85.14 | 69.31 | 52.68 |
| FedRep [8] | 87.07 | 82.40 | 65.71 | 50.36 |
| FedRoD [4] | 87.61 | 80.36 | 63.90 | 51.42 |
| FedBABU [37] | 83.15 | 76.33 | 55.91 | 50.19 |
| ***Efficient Federated Learning*** | | | | |
| FedVPT [21] | 89.39 | 85.11 | 55.49 | 45.26 |
| FedVPT-D [21] | 89.56 | 85.43 | 66.91 | 50.25 |
| pFedPG (Ours) | **90.08** | **87.57** | **70.96** | **55.91** |

- Evaluations in *domain shift* & *imbalanced class distribution* verify that our pFedPG outperforms existing FL methods (***+ 9%*** Avg. over FedAvg on DomainNet).
- Our pFedPG exhibits sufficient training efficiency (***0.01%*** parameters).

## Ablation Study & Analysis

| Module | Method | Office-Caltech10 | DomainNet | CIFAR-10 | CIFAR-100 |
|---|---|---|---|---|---|
| Prompt generation | FedVPT | 94.29 | 64.16 | 89.39 | 55.49 |
| | $\mathbf{P}_{base}$ | 93.16 | 64.87 | 88.23 | 66.89 |
| Architecture of $G$ | MLP [44] | 94.96 | 63.33 | 87.47 | 66.73 |
| | AdaIN [20] | 95.72 | 70.08 | 89.77 | 69.44 |
| | **pFedPG** | **96.81** | **71.64** | **90.08** | **70.96** |

| *K* | Office-Caltech10 | DomainNet | CIFAR-10 | CIFAR-100 |
|---|---|---|---|---|
| 1 | 96.09 | 70.27 | 86.14 | 55.77 |
| 5 | 96.77 | 70.53 | 87.41 | 55.79 |
| 10 | **96.81** | **71.64** | **87.57** | **55.91** |
| 50 | 95.10 | 69.55 | 85.63 | 54.52 |
| 100 | 94.53 | 68.79 | 85.02 | 53.61 |
| 200 | 94.46 | 66.83 | 83.53 | 52.34 |

## Conclusion

- We propose pFedPG to enable efficient model personalization under heterogeneous clients' data.
- We effectively exploit personalized optimization directions and produce client-specific prompts for updating each client model.