Federated Learning via Meta-Variational Dropout

Insu Jeon, Minui Hong, Junhyeog Yun, Gunhee Kim

Thirty-seventh Conference on Neural Information Processing Systems, Dec 10th, 2023

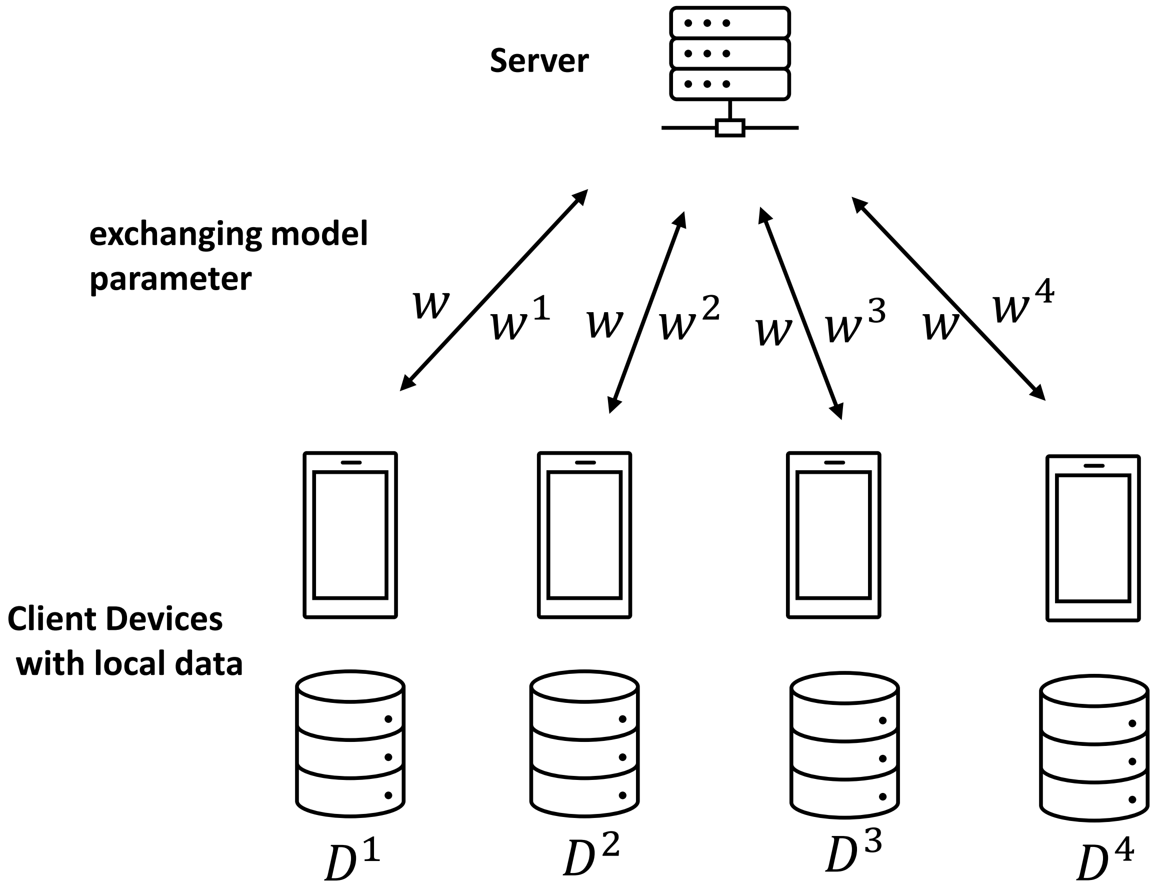
Seoul National University

VERI LUX TAS MEA

SEOUL NATIONAL UNIV.
VISION & LEARNING

NEURAL INFORMATION
PROCESSING SYSTEMS

Federated Learning



- **Federated Learning** (FL) is a method for training AI models from distributed client devices.

- Models are trained on each client, and only parameters are shared with the server to improve the global model.

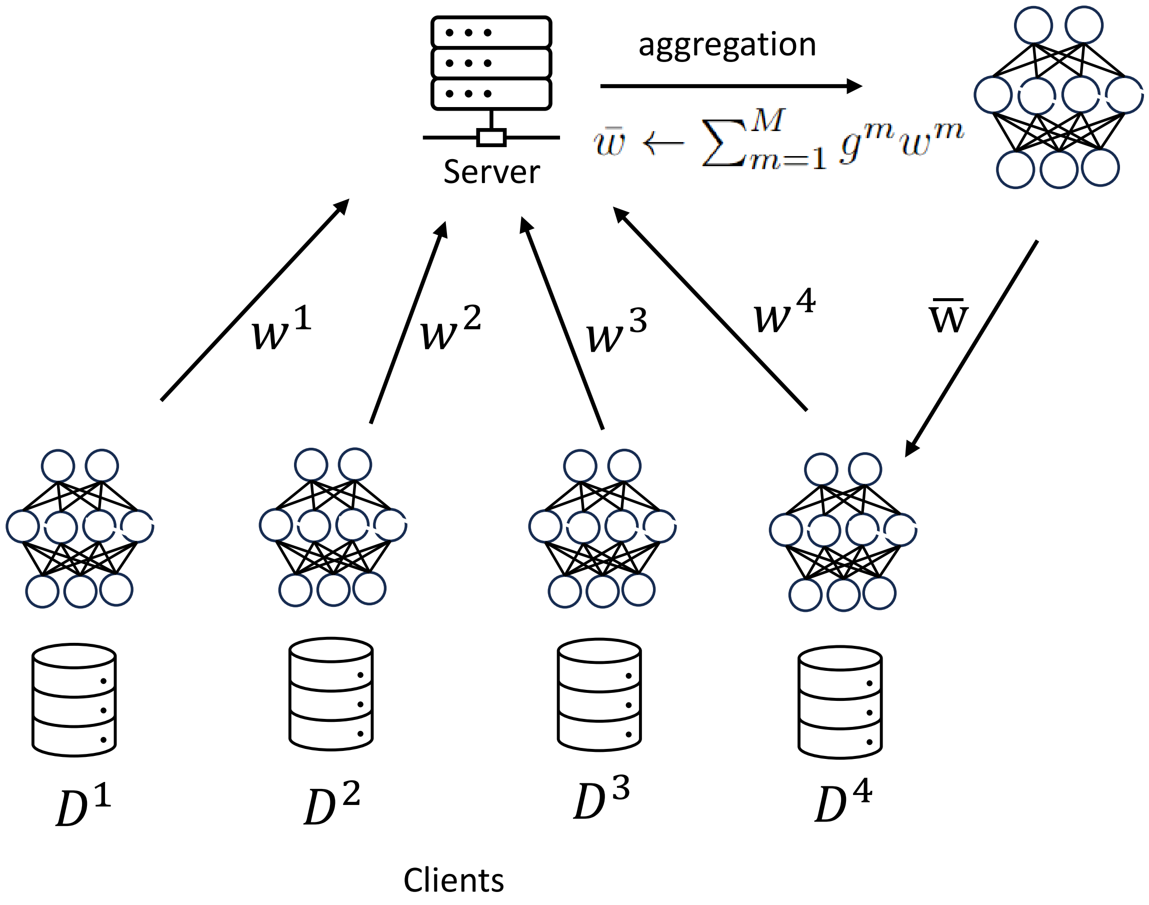
- Because FL do not exchange the client data, an organization can collaborate without exposing sensitive information and compromising data privacy.

FL Objective

$$\text{Server: } \min_{w} \mathcal{J}(w) = \sum_{m=1}^{M} g^m \mathcal{J}^m(w), \qquad \text{Client: } \mathcal{J}^m(w) = \frac{1}{|\mathcal{D}^m|} \sum_{i} \ell(x_i^m, y_i^m; w)$$

- In **FL objective**, the global objective $\mathcal{J}(w)$ at server can be represented as a linear combination of local objectives $\mathcal{J}^m(w)$ at each client. w is the model parameter.
- Given each client has a data D^m, the local object is usually defined as the negative log-likelihood $\ell(D^m; w) = -\log p(y^m | x^m, w)$ on the m-th client's dataset.
- g^m is a weight proportional to the size of the local dataset (e.g., $|D^m|/|D|$)

FedAvg



- **FedAvg** [1] is a basic FL approach to find an optimal model, proposed by McMahan et al. in 2017.

$$\text{Server: } \bar{w} \leftarrow \sum_{m=1}^{M} g^m w^m$$

$$\text{Client: } \bar{w} \xrightarrow{\text{SGD}} w^m \quad (m = 1, \ldots, M)$$

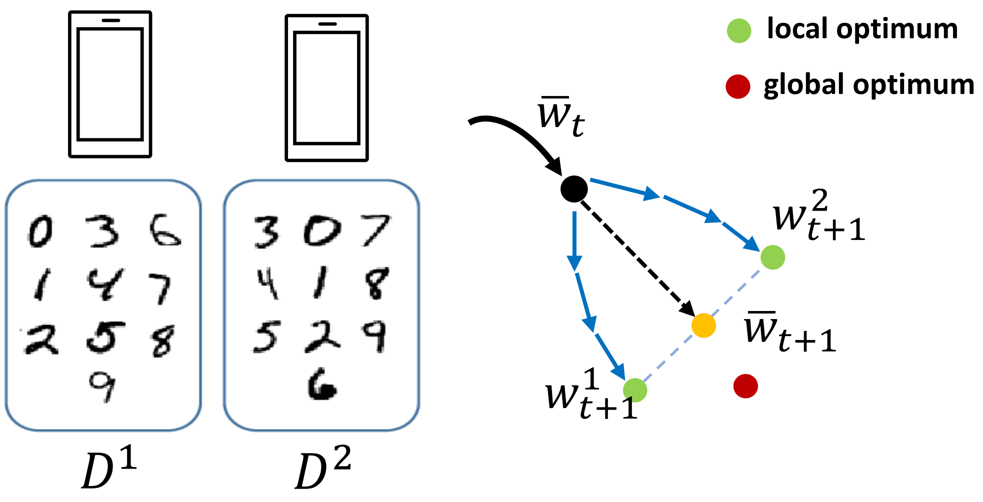
- It iteratively updates the global model parameter $\overline{\mathrm{w}}$ with the average of local parameter w^{m} weighted by each client's data size.

- However, conventional FL has difficulties in the real-world application.

[1] McMahan et al., Communication-Efficient Learning of Deep Networks from Decentralized Data, AISTAT, 2017

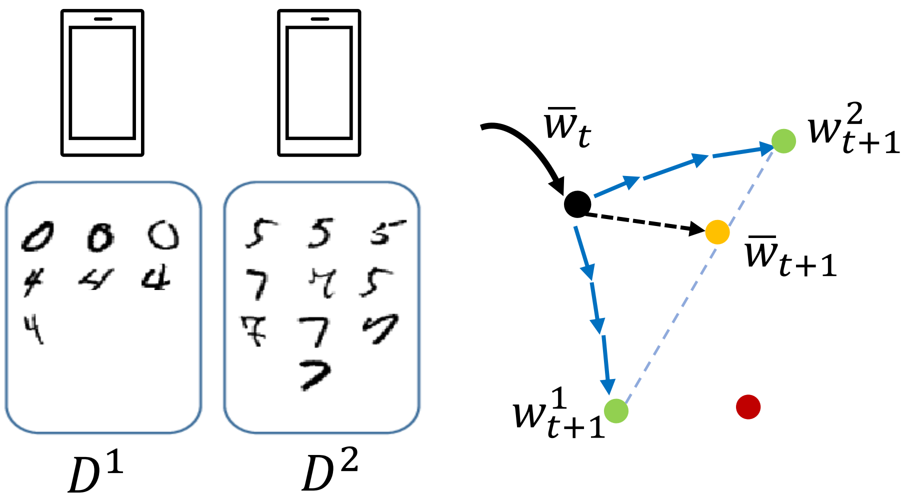
Challenge in FL with Non-IID clients

IID data



IID clients: The conventional FL algorithm, (e.g., FedAVG) could convergence well when the data from different clients is independently and identically distributed (IID).

Non-IID data



Non-IID data: Due to the differences in preferences, locations, and usage habits of clients, the private data are usually non-IID. In this case, the local model learned from each client can diverge, and thus learning an optimal global model could fail.

Other Challenges in FL

- **Limited or Unbalanced data:** When the data available on each local device is limited, the model overfitting occurs, resulting in poor generalization to unseen clients.

- **Sparse participation**: In practice, the total number of clients M can be extremely large, while communication between the server and the clients can be intermittent or unreliable. This creates the challenge of inconsistent training due to a small subset of participating clients in each round of communication.

- **Communication cost:** FL optimization requires frequent communication between local devices and the central server to exchange model parameters. This process is slow and could introduce additional privacy concerns. Therefore, reducing model size is also an important area of research.

Motivation

- Our research goal was to mitigate these **major challenges in FL** (such as non-IID clients, data imbalance, few-shot learning, sparse client participation, and communication costs) simultaneously.

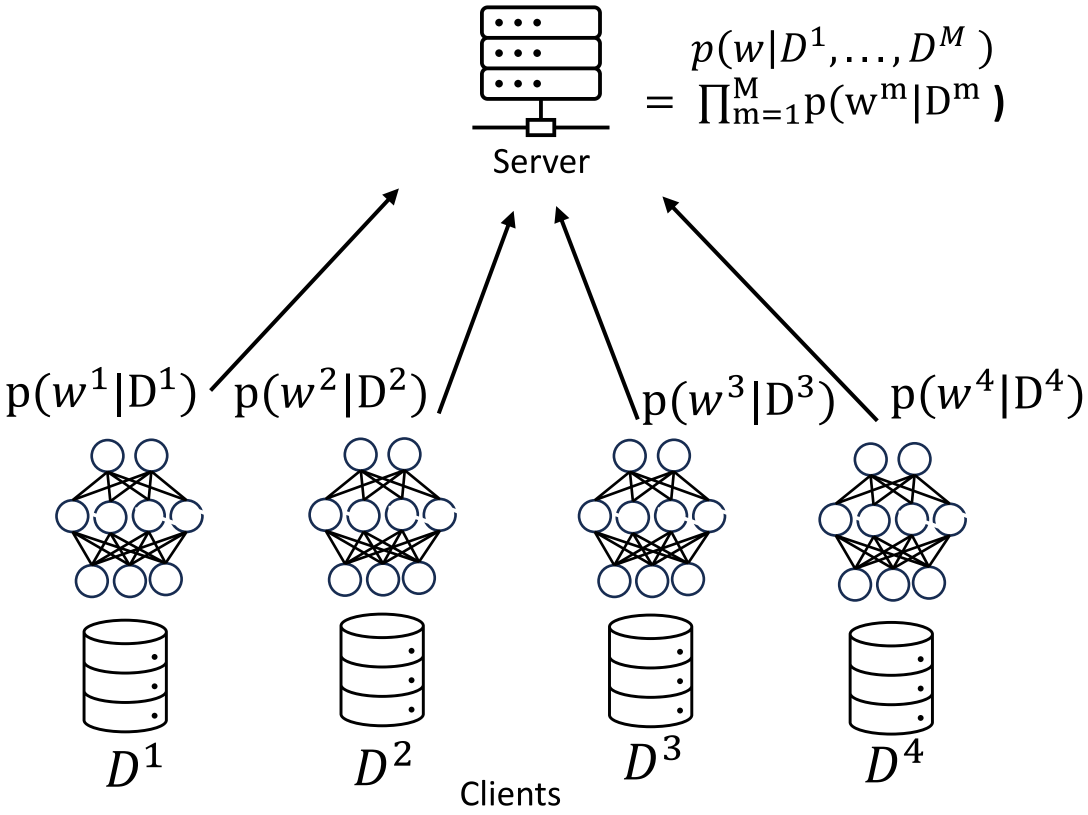
- How do we address these issues?

  - We introduce a new Bayesian FL framework, called **Meta-Variational Dropout**.

  - This framework includes,

  1) Variational Inference for FL
  2) Conditional Variational Dropout approach based on Hypernetwork
  3) Sparse prior, enabling model compression
  4) Model aggregation rule based on parameter uncertainty

Bayesian Federated Learning



Bayesian focuses on learning local posteriors $p(w^m|D^m)$ in each client's device, then we utilizes those local posteriors to infer the global posterior $p(w|D) = p\left(w\middle|D^1, \dots, D^M\right)$.

In other words, Bayesian approach employes a distribution over the model instead of a fixed-point estimation of model parameter.

Variational Inference for FL

❖ Given M clients, each of which has a dataset $D^m = \{(x_i^m, y_i^m)\}_{i=1}^{|D^m|}$, **an evidence lower-bound (ELBO) objective** over all the distributed client's dataset can be defined as

$$\max_{\phi} \mathcal{L}_{ELBO}(\phi) = \sum_{m=1}^{M} g^m \{\mathbb{E}_{q(w^m;\phi)}[\log p(y^m|x^m, w^m)] - KL(q(w^m;\phi)||p(w^t))\}$$

- $q(w^m\,;\,\emptyset)$ is a variational posterior (or probability distribution) over w^m
- $p(y^m|x^m, w^m)$ represents a likelihood model based on a neural network (NN) on each m-th client's data, and w^m is a client-specific NN parameter
- $p(w^m)$ is a prior distribution that acts as a regularizer for the $q(w^m\,;\,\emptyset)$
- g^m is the local weight is proportional to the size of the local dataset (e.g., $|D^m|/|D|$)
- $q(w^m\,;\phi) \approx p(w^m \mid D^m)$.

Posterior model

- **MetaVD extends the posterior model of Variational Dropout (VD)** [2] by introducing a hypernetwork h_{φ} (and embedding e^m) to predict a client-specific dropout variable α^m.

$$q(w^m; \phi = (\theta, \psi, e^m)) = \prod_{k=1}^{K} \mathcal{N}(w_k^m | \theta_k, \alpha_k^m \theta_k^2) \text{ where } \alpha^m = h_{\psi}(e^m)$$

- The θ is (a typically fixed) neural network parameter that is a globally shared across clients.

- The dropout technique switches off the neurons of a deep learning model with a certain probability to sample a neural network with a different structure each time.

- This has a similar effect to ensemble learning and improves generalization performance by reducing the dependence between neurons and prevent model overfitting during training.

[2] Diederik P. Kingma te al., Variational Dropout and Local Reparameterization Trick. NeurIPS 2015

Prior

❖ **MetaVD adopted the hierarchical prior** [3] to enforce a sparsity on the global weight. The KL regularization terms in the ELBO for FL is defined as

$$\mathrm{KL}(q(w^m;\phi)||p(w^m)) = \sum_{k=1}^{K} 0.5\log(1+(\alpha_k^m)^{-1})$$

- (1) The two-level structure in a hierarchical system can generate a much more complex distribution, expanding the potential solution spaces for variety of different clients' models in the FL environment.
- (2) This has been proven effectiveness in network regularization and scarification.
- The same hierarchical prior is uniformly applied across all 1...M clients to ensure the Bayesian posterior aggregation rule.

[3] Yuhang Liu et al., Variational Bayesian Dropout with a Hierarchical Prior, CVPR 2019

Bayesian Posterior Aggregation

- To update the global NN parameter, we first assume the Bayesian posterior aggregation rule,

$$p(w|\mathcal{D}) \propto \prod_{m=1}^{M} p(w^m|\mathcal{D}^m)$$

- Since we have approximated the conditional posterior as Gaussian Dropout, we can denote the global posterior as a product of local Gaussians,

$$\mathcal{N}(w|\theta_*^{\text{agg}}, \cdot) \approx \prod_{m=1}^{M} \mathcal{N}(w^m|\theta_*^m, \alpha_*^m(\theta_*^m)^2)$$

❖ MetaVD derives an exact aggregation rule to compute a maximum a posterior (MAP) solution of the parameter as

$$\theta_*^{\text{agg}} = \frac{1}{M}\sum_m r^m \theta_*^m \text{ where } r^m = \frac{g^m(\alpha_*^m(\theta_*^m)^2)^{-1}}{\sum_m g^m(\alpha_*^m(\theta_*^m)^2)^{-1}}$$

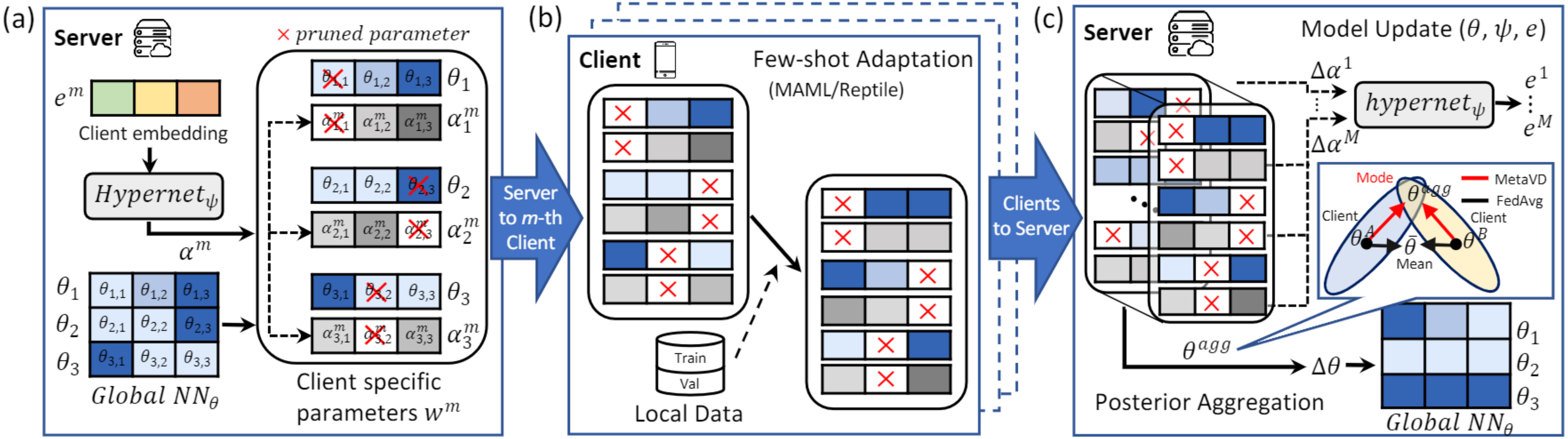
Combination with MAML/Reptile

- **MetaVD** is compatible with several meta-learning based PFL algorithms such as Reptile, MAML, PerFedAvg.

- Unlike conventional meta-learning algorithms, MetaVD changes the mode of initialization parameters for each client.

- MetaVD prevents overfitting of local adaptation in meta-learning based based PFL approach.

Overall FL with MetaVD



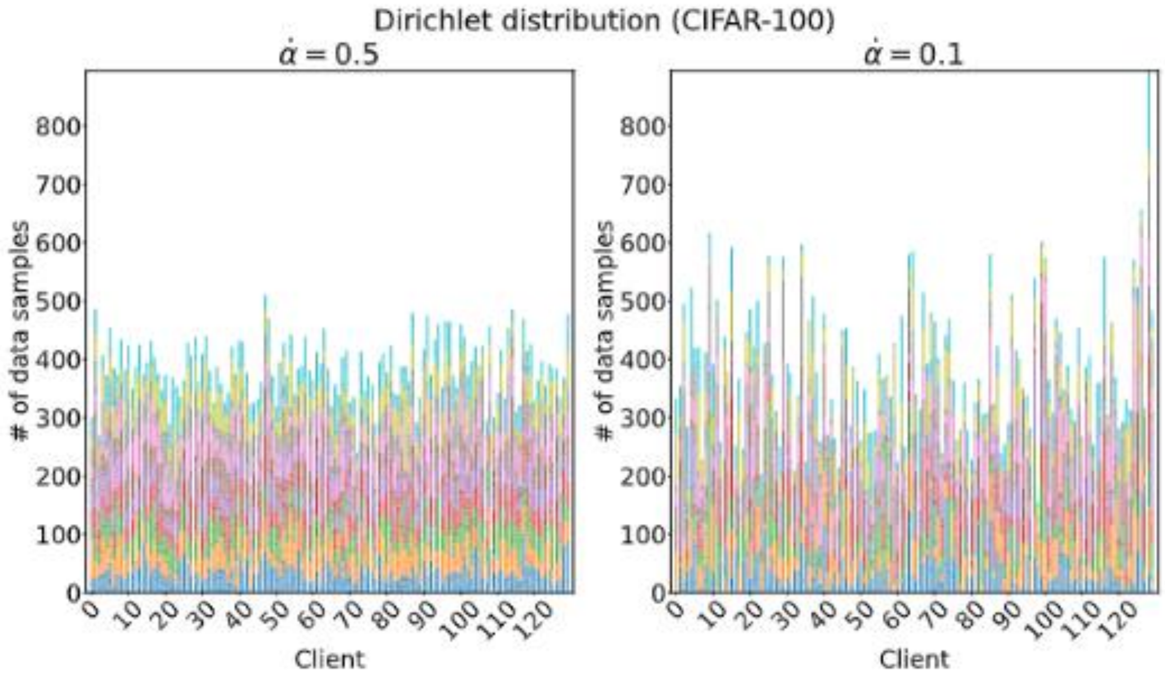
(a) Initialization at Server: (1) MetaVD estimates client specific dropout variable α^m using a hypernetwork h_φ and embedding vector e^m. (2) Apply dropout to Global NNs' weight θ and send $\{\theta, \alpha^m\}$ to each client -> **reduce communication cost through dropout pruning**.

(b) Optimization at each Client: A client receives personalized model, and train the global parameter and dropout rates using the local data, then send them back to server. (Here, **we can also utilize the optimization based meta-learning approach**, **Reptile or MAML**).

(c) Aggregation at Server: When merging models learned from multiple local clients, update the global weight θ inversely proportional to the dropout rate learned from each client (merging models reflecting the uncertainty of weight) -> lead to **better optimum.**

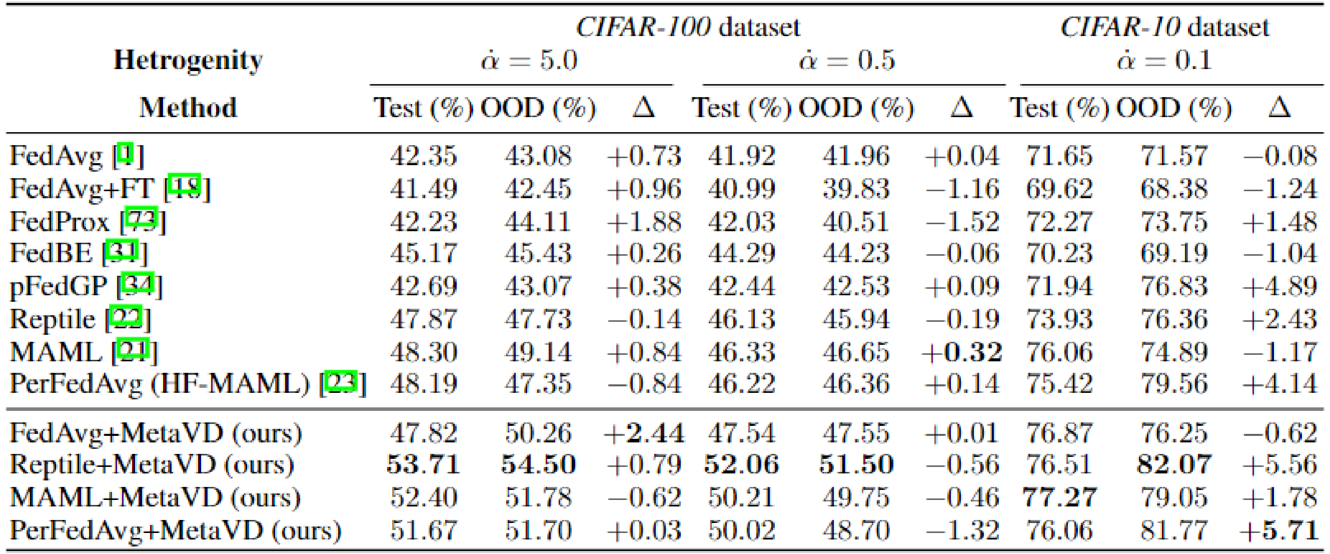
Experiment

Classification with Non-IID data



Dirichlet Distribution was used to control the diversity level of FL client data distribution (e.g., size, degree of non-IID of class, etc.).

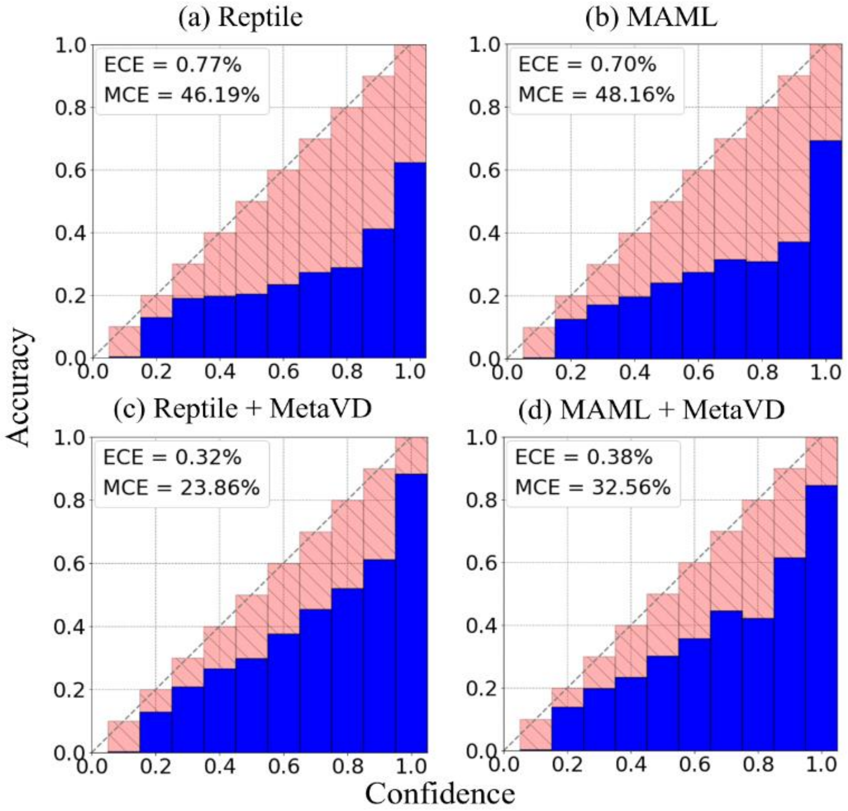
Classification with Non-IID data



	CIFAR-100 dataset						*CIFAR-10* dataset		
Hetrogenity	$\dot{\alpha} = 5.0$			$\dot{\alpha} = 0.5$			$\dot{\alpha} = 0.1$		
Method	Test (%)	OOD (%)	Δ	Test (%)	OOD (%)	Δ	Test (%)	OOD (%)	Δ
FedAvg [1]	42.35	43.08	+0.73	41.92	41.96	+0.04	71.65	71.57	−0.08
FedAvg+FT [18]	41.49	42.45	+0.96	40.99	39.83	−1.16	69.62	68.38	−1.24
FedProx [73]	42.23	44.11	+1.88	42.03	40.51	−1.52	72.27	73.75	+1.48
FedBE [31]	45.17	45.43	+0.26	44.29	44.23	−0.06	70.23	69.19	−1.04
pFedGP [34]	42.69	43.07	+0.38	42.44	42.53	+0.09	71.94	76.83	+4.89
Reptile [22]	47.87	47.73	−0.14	46.13	45.94	−0.19	73.93	76.36	+2.43
MAML [21]	48.30	49.14	+0.84	46.33	46.65	**+0.32**	76.06	74.89	−1.17
PerFedAvg (HF-MAML) [23]	48.19	47.35	−0.84	46.22	46.36	+0.14	75.42	79.56	+4.14
FedAvg+MetaVD (ours)	47.82	50.26	**+2.44**	47.54	47.55	+0.01	76.87	76.25	−0.62
Reptile+MetaVD (ours)	**53.71**	**54.50**	+0.79	**52.06**	**51.50**	−0.56	76.51	**82.07**	+5.56
MAML+MetaVD (ours)	52.40	51.78	−0.62	50.21	49.75	−0.46	**77.27**	79.05	+1.78
PerFedAvg+MetaVD (ours)	51.67	51.70	+0.03	50.02	48.70	−1.32	76.06	81.77	**+5.71**

- Tested at different Non-IID levels utilizing FL Benchmark datasets (CIFAR-100 and CIFAR-10).
- When the MetaVD is combined with traditional FL algorithms (FedAvg) or meta-learning based algorithms (Reptile, MAML, PerFedAvg), it shows a significant improvement in classification accuracy.
- In addition to the test data, accuracy was significantly improved for out-of-distribution (OOD) clients that were never seen during training.

Uncertainty Calibration

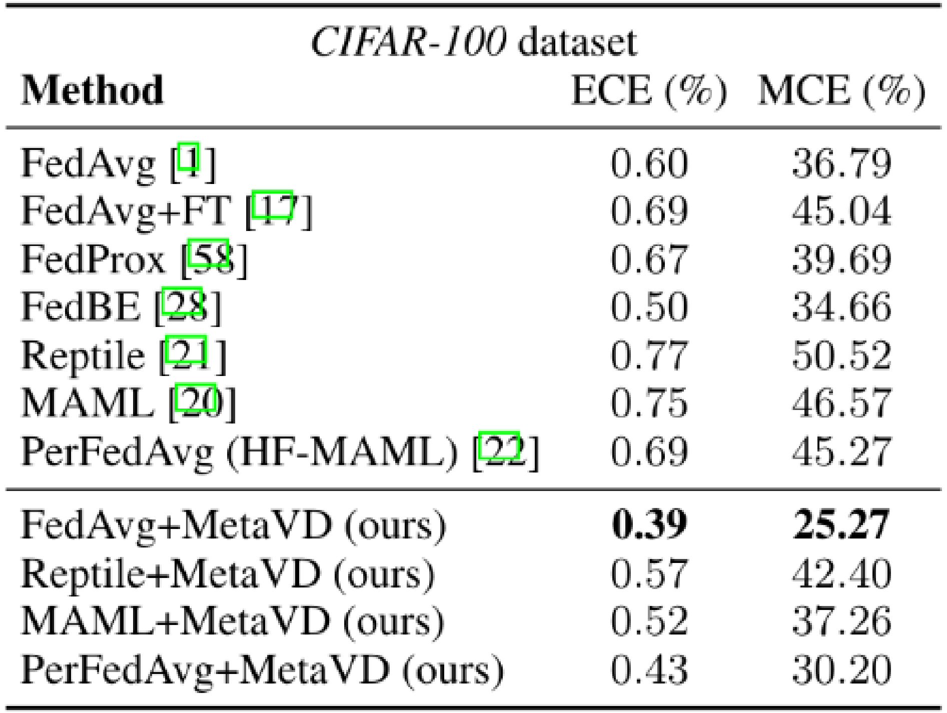


- To measure how well the FL model's predictions match the actual probability values, we analyzed the reliability diagram for the CIFAR-100 dataset.

- The **reliability diagram** measures the proportion of true-positive samples in each probability interval. The lower the bias of the model, the closer to the diagonal line it is drawn.

- As a result of the experiment, we observed that **the reliability of most baselines** such as Reptile and MAML **improved when MetaVD was applied**.

Uncertainty Calibration (2)

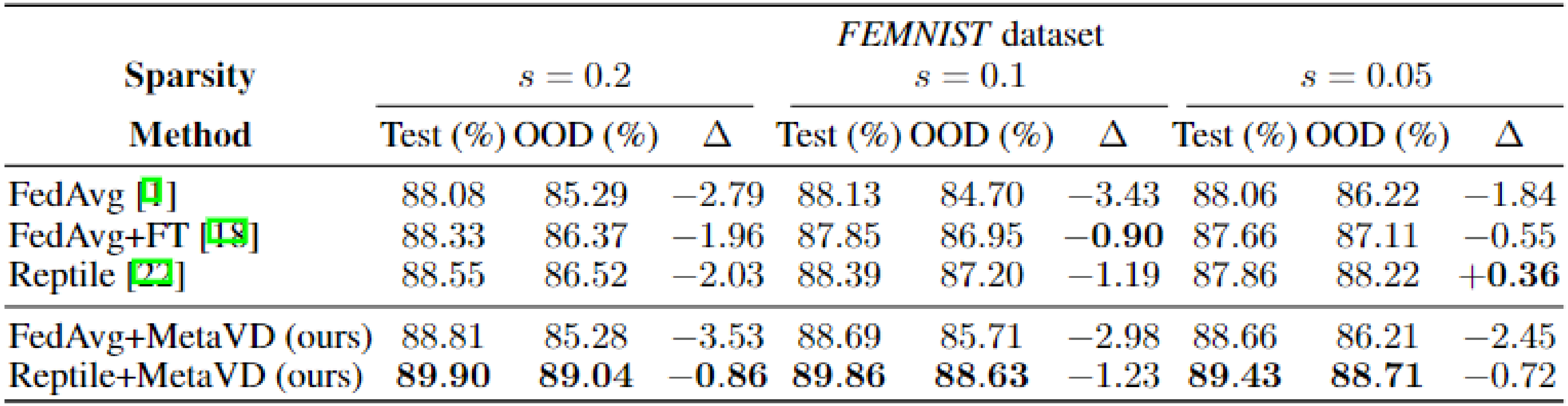


CIFAR-100 dataset		
Method	ECE (%)	MCE (%)
FedAvg [1]	0.60	36.79
FedAvg+FT [17]	0.69	45.04
FedProx [58]	0.67	39.69
FedBE [28]	0.50	34.66
Reptile [21]	0.77	50.52
MAML [20]	0.75	46.57
PerFedAvg (HF-MAML) [22]	0.69	45.27
FedAvg+MetaVD (ours)	**0.39**	**25.27**
Reptile+MetaVD (ours)	0.57	42.40
MAML+MetaVD (ours)	0.52	37.26
PerFedAvg+MetaVD (ours)	0.43	30.20

- ECE score represents the average difference between the model predictions and the actual probability, while MCE score represents the maximum difference.

- We observe a much lower error when applying MetaVD. This indicates that **MetaVD improves the reliability of model predictions.**

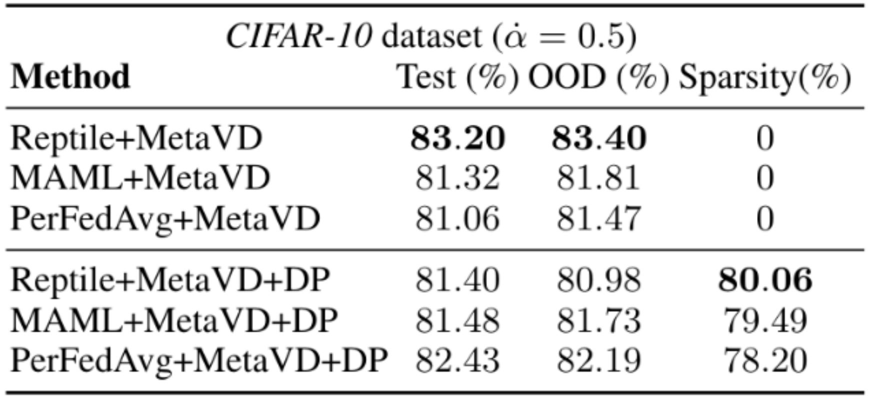
Client Participation Degrees



	FEMNIST dataset								
Sparsity	$s = 0.2$			$s = 0.1$			$s = 0.05$		
Method	Test (%)	OOD (%)	Δ	Test (%)	OOD (%)	Δ	Test (%)	OOD (%)	Δ
FedAvg [1]	88.08	85.29	-2.79	88.13	84.70	-3.43	88.06	86.22	-1.84
FedAvg+FT [18]	88.33	86.37	-1.96	87.85	86.95	$\mathbf{-0.90}$	87.66	87.11	-0.55
Reptile [22]	88.55	86.52	-2.03	88.39	87.20	-1.19	87.86	88.22	$\mathbf{+0.36}$
FedAvg+MetaVD (ours)	88.81	85.28	-3.53	88.69	85.71	-2.98	88.66	86.21	-2.45
Reptile+MetaVD (ours)	**89.90**	**89.04**	$\mathbf{-0.86}$	**89.86**	**88.63**	-1.23	**89.43**	**88.71**	-0.72

- We used the FEMNIST dataset to measure the prediction accuracy of the model by adjusting the percentage (%) of clients participating in training (s = 0.2, 0.1, 0.05).
- In all cases, the MetaVD improved prediction results of baseline even with the sparse client participation degrees.

Model Compression



CIFAR-10 dataset ($\dot{\alpha} = 0.5$)			
Method	Test (%)	OOD (%)	Sparsity(%)
Reptile+MetaVD	**83.20**	**83.40**	0
MAML+MetaVD	81.32	81.81	0
PerFedAvg+MetaVD	81.06	81.47	0
Reptile+MetaVD+DP	81.40	80.98	**80.06**
MAML+MetaVD+DP	81.48	81.73	79.49
PerFedAvg+MetaVD+DP	82.43	82.19	78.20

- When applying MetaVD, the dropout probability per model weight is learned.
- When learning FL, weights with dropout rates above 0.8 are pruned and not exchanged between server and client (+DP), and the final prediction performance is not significantly reduced.

Conclusion

- **MetaVD** is a novel Bayesian meta-learning approach to FL extending Variational Dropout.
- **Hypernetwork Utilization**: MetaVD predicts dropout rates for each NN parameter of each client, supporting model personalization and adaptation in FL with non-IID and limited data.
- **Uncertainty in PFL aggregation**: Variational dropout uncertainty is firstly utilized as a principled Bayesian aggregation rule in PFL, improving training convergence and prevent model overfitting.
- **OOD Performance**: MetaVD has been tested on various FL scenarios. It prevents model over-fitting and significantly improves prediction performance over the OOD clients. In addition, it has the effect of compressing the model, which reduces communication costs.
- **Generic Compatibility**: MetaVD is a highly versatile methodology. It works with any existing meta-learning algorithms to avoid overfitting. The application in broader domains (NLP, RL, etc.) is an interesting future work.