FedAvP: Augment Local Data via Shared Policy in Federated Learning

Minui Hong
Seoul National University

Junhyeog Yun
Seoul National University

Insu Jeon
Seoul National University

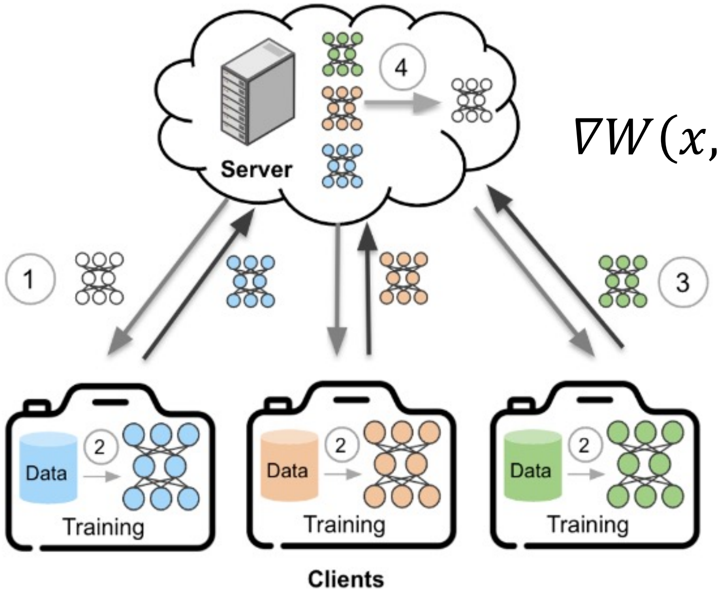
Gunhee Kim
Seoul National University

Motivation

- Federated Data Augmentation Aim to increase **the diversity and volume of data** available at each client, thereby improving overall performance of the federated models.
- Sharing input-level or feature-level information can **raise privacy concerns**.

Background : Reconstruction Attack

- Assume we have a given gradient $\nabla W(x, y)$, then we **can optimize for a dummy data** and label pair (x', y') by minimizing the following objective[1]:



$\nabla W(x, y)$

$$x^*, y^* = \underset{x',y'}{\operatorname{argmin}} ||\boldsymbol{\nabla W(x, y)} - \nabla W(x', y')||$$

Reconstruct Data

[Fig] Federated learning training process, https://ai.sony/blog/blog-032/
[1] Zhu, L., et al. "Deep leakage from gradients", NeurIPS 2019

Motivation

Yoon et al., ICLR'21

$$x^*, y^* = \underset{x', y'}{\operatorname{argmin}} [(1-\alpha) \cdot \left(1 - \ell\left(\nabla W(x,y), \nabla W(x,' y')\right)\right) + \alpha \cdot ||x' - x_{mean}||]$$

Mean Image

Zhu, Z., et al., PMLR'21

$$x^*, y^* = \underset{x', y'}{\operatorname{argmin}} [(1-\alpha-\beta) \cdot \left(1 - \ell\left(\nabla W(x,y), \nabla W(x,' y')\right)\right) + \alpha \cdot \left||c - y'|\right| + \beta \cdot (1 - \ell(W^p(z|z\sim G(y')), W(x'))]$$

Label & Generator

Zhou & Konukoglu, ICLR'23

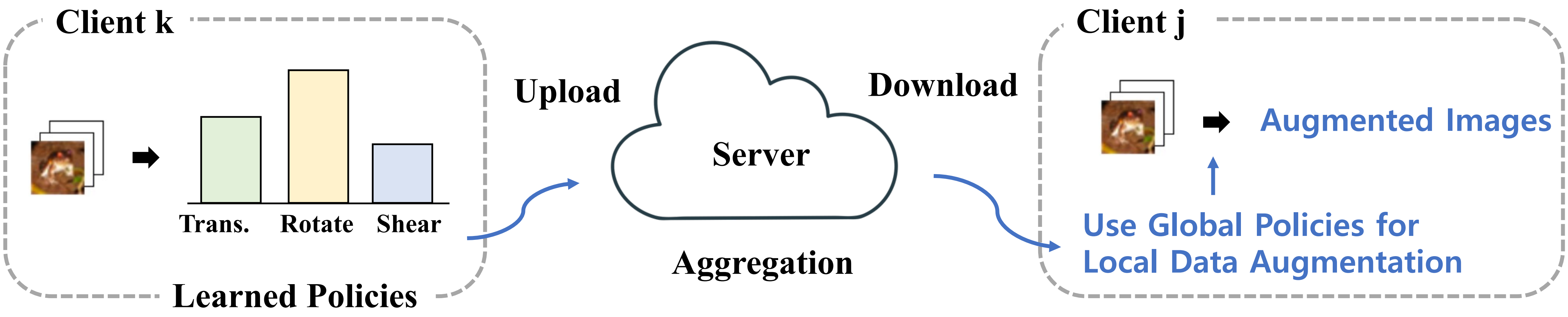
$$x^*, y^* = \underset{x', y'}{\operatorname{argmin}} [(1-\alpha-\beta) \cdot \left(1 - \ell\left(\nabla W(x,y), \nabla W(x,' y')\right)\right) + \alpha \cdot \mathbb{E}_k \left|\left|\bar{\mu}^k - \mu^{k'}\right|\right| + \beta \cdot \mathbb{E}_k \left|\left|\bar{\sigma}^k - \sigma^{k'}\right|\right|]$$

Feature Statistics

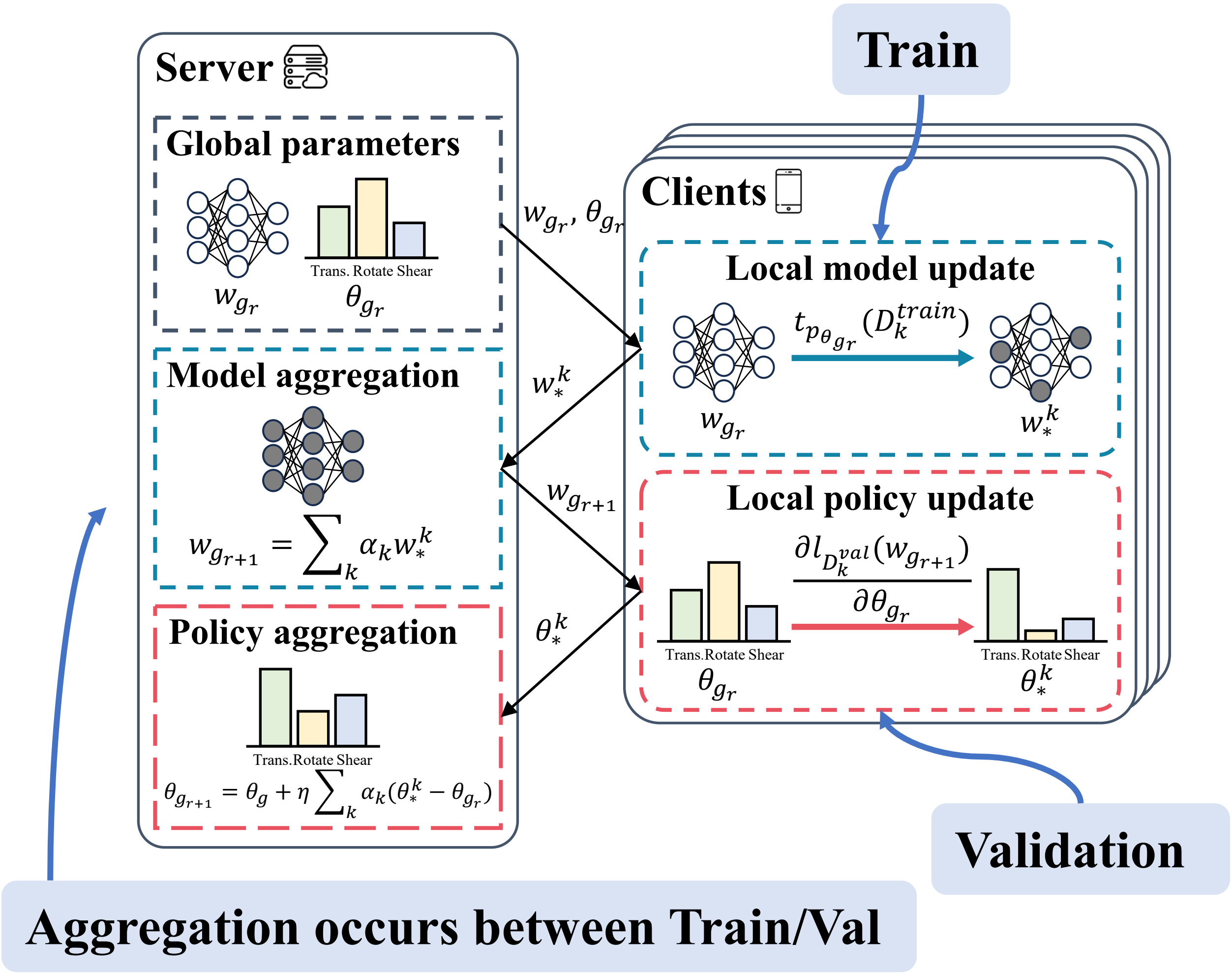
[1] Yoon, Tehrim, et al. "FedMix: Approximation of Mixup under Mean Augmented Federated Learning.", ICLR 2021
[2] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. "Data-free knowledge distillation for heterogeneous federated learning.", PMLR 2021
[3] Zhou & Konukoglu. "FedFA: Federated Feature Augmentation.", ICLR 2024

Our Idea

- Instead of directly sharing raw data between clients, we share Data **Augmentation Policies**.
- Augmentation Policies include the types and intensities of image transformations such as translation, shear, and flipping.



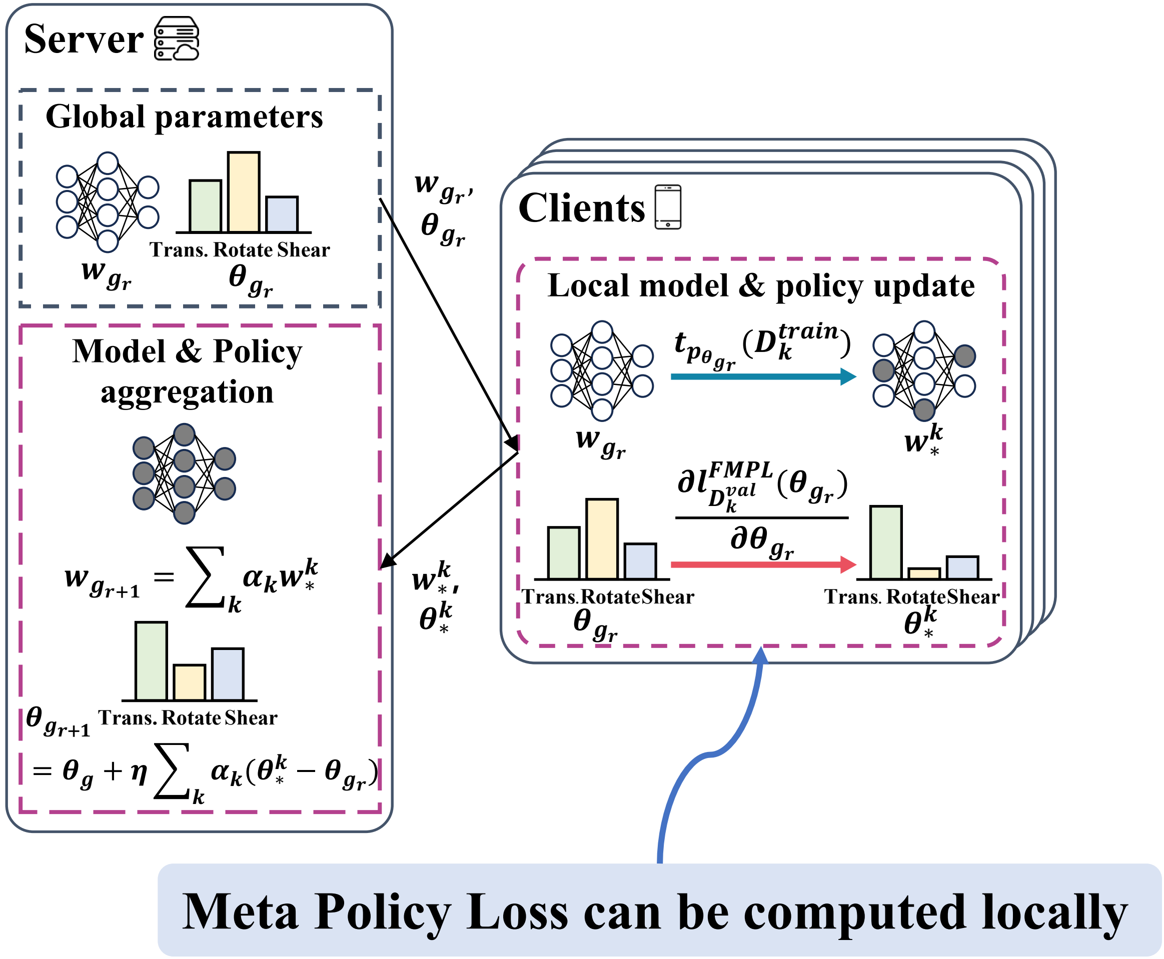
Policy Optimization



(a) Our Federated Meta Policy Loss(FMPL)

- One round of local training and aggregation is considered an **inner step**.
- Validation on clients after each global model update servers as the **outer step**.

Policy Optimization



(b) First-order Approximation of FMPL

- Accessing other clients' gradients raises **privacy and communication** concerns.
- To address this, we apply **a first-order approximation.**

Policy Optimization

- We apply a first-order approximation via Taylor expansion, **reducing both privacy risks and communication overhead.**

Consider the federated meta-policy loss derived from the updated weight w_n^k for client k at step n using a first-order Taylor expansion:

$$\ell_{D_k^{val}}\left(w_{g_{r+1}}\right) \approx \ell_{D_k^{val}}\left(w_n^k\right) + \nabla \ell_{D_k^{val}}\left(w_n^k\right)^T \left(w_{g_r} - w_n^k\right).$$

When computing the policy gradient of the loss with respect to θ_{n-1}^k, the first-order gradient approximation is

$$-\alpha_k \cdot lr \frac{\partial\left(\nabla \ell_{D_k^{val}}\left(w_n^k\right)^T \nabla \ell_{tp_{\theta_{n-1}^k}\left(D_{k,n-1}^{train}\right)}\left(w_{n-1}^k\right)\right)}{\partial \theta_{n-1}^k},$$

where $w_n^k = w_{g_r} - lr \cdot g_{w_0^k}^{aug} - \cdots - lr \cdot g_{w_{n-1}^k}^{aug}$ and α_k is a coefficient proportional to the client's data size.

Policy Optimization

- We update our policy as done in Reptile[1]. Our algorithm **allows for rapid adaptation of a personalized policy** by each client.

We train the policy neural network by increasing the dot-product between policy gradients on each client as follows:

$$\theta_{g_{r+1}} \approx \theta_{g_r} - \eta\lambda \frac{\partial}{\partial \theta_0^k} \mathbb{E}\left[\sum_{j=1}^{n} L_{k,j} - \frac{\lambda}{2} \sum_{j=0}^{n} \sum_{s=0}^{j-1} \langle \nabla L_{k,j} \cdot \nabla L_{k,s} \rangle \right],$$

where $L_{k,j} = \ell_{D_{k,j}^{val}}^{FMLPL}(\theta_0^k)$ *is the federated meta-policy loss computed on the client k's j-th validation data batch using the global policy parameters* θ_0^k.

[1] Nichol, Alex, Joshua Achiam, and John Schulman. "On first-order meta-learning algorithms." arXiv 2018.

Experiments

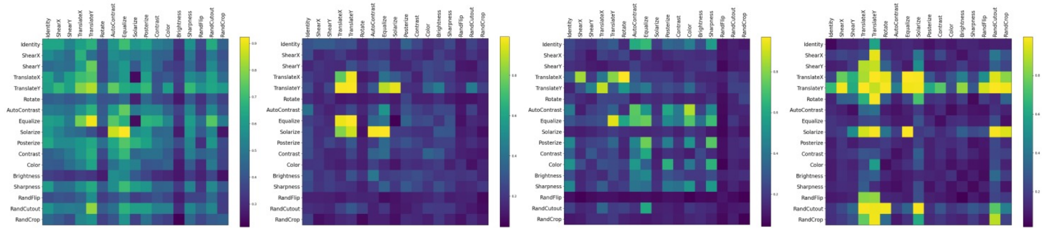
- Non-IID Classification Results.

Dataset / Method		CIFAR-100 $\alpha = 5.0$ Test (%)	CIFAR-100 $\alpha = 0.1$ Test (%)	CIFAR-10 $\alpha = 5.0$ Test (%)	SVHN $\alpha = 0.1$ Test (%)	FEMNIST Test (%)
FedAvg	+ Default	40.05	37.34	79.76	85.58	80.65
	+ RandAugment	47.29	43.60	82.82	84.84	79.40
	+ TrivialAugment	46.61	42.16	82.00	83.36	79.01
FedProx	+ Default	40.57	37.71	80.64	86.79	81.45
	+ RandAugment	45.97	41.39	82.56	85.52	77.11
	+ TrivialAugment	46.61	41.81	81.83	84.11	79.67
FedDyn	+ Default	42.09	38.52	80.36	87.60	80.47
	+ RandAugment	45.70	42.24	82.51	81.47	77.64
	+ TrivialAugment	46.83	41.10	82.03	83.41	79.31
FedExP	+ Default	42.76	38.28	80.64	86.66	81.45
	+ RandAugment	46.13	42.23	82.86	84.63	79.69
	+ TrivialAugment	48.55	42.09	82.51	83.72	80.20
FedGen	+ Default	42.14	38.27	80.23	86.79	81.86
	+ RandAugment	47.11	43.10	81.90	84.39	79.34
	+ TrivialAugment	47.71	40.76	82.58	83.23	77.35
FedMix	+ Default	40.26	38.69	80.99	86.02	81.63
	+ RandAugment	46.69	43.00	83.08	83.44	79.46
	+ TrivialAugment	46.64	42.63	81.83	82.34	77.84
FedFA	+ Default	43.70	41.21	82.61	87.33	81.13
	+ RandAugment	48.86	43.44	82.44	81.32	78.71
	+ TrivialAugment	47.86	43.45	80.12	78.62	78.96
FedAvP	(W/ Local Policy)	49.04	43.86	83.64	87.05	83.94
FedAvP	(Fast Update)	49.97 ($\pm$0.04)	45.08 ($\pm$0.01)	83.55 ($\pm$0.06)	87.86 ($\pm$1.53)	84.47 ($\pm$0.006)
FedAvP		50.47 ($\pm$0.03)	45.96 ($\pm$0.01)	83.78 ($\pm$0.004)	89.81 ($\pm$1.55)	84.27 ($\pm$0.07)

- Results with a larger model.

Method	$R = 100$ Test (%)	$R = 100$ OOD (%)	$R = 300$ Test (%)	$R = 300$ OOD (%)	$R = 500$ Test (%)	$R = 500$ OOD (%)
FedAvg+Default	76.01	74.18	83.92	82.59	90.38	91.64
FedAvg+RandAugment	59.30	53.52	81.04	77.74	89.75	89.96
FedAvg+TrivialAugment	44.74	40.79	78.83	75.59	89.51	89.80
FedExP+Default	84.89	84.97	87.17	87.44	90.03	90.72
FedExP+RandAugment	74.44	71.87	87.56	85.31	88.20	88.92
FedExP+TrivialAugment	43.56	40.47	83.26	81.51	88.07	88.68
FedFA+Default	83.19	83.10	88.99	89.45	91.18	92.03
FedFA+RandAugment	62.62	63.74	86.77	86.23	90.92	91.97
FedFA+TrivialAugment	8.477	10.63	68.42	70.29	86.87	88.09
FedAvP (Fast Update)	**86.14**	**87.24**	**91.56**	**92.13**	**93.85**	**93.34**

- Visualization of global policies.



(a) CIFAR-100, $\alpha = 5.0$ (b) CIFAR-100, $\alpha = 0.1$ (c) SVHN, $\alpha = 0.1$ (d) FEMNIST

Experiments

- Reconstruction attack results.

Metric / Method	PSNR Client(S)	PSNR Client(L)	Accuracy Test(%)
FedAvg	10.88	11.36	37.34
FedGen	8.86	9.27	38.27
FedMix	10.27	10.48	38.69
FedFA	10.86	11.82	41.21
FedAvP	8.72	9.25	**45.96**
FedGen + label + generator	9.21	9.81	38.27
FedMix + input	11.89	12.40	38.69
FedFA + feature	12.11	12.87	41.21
FedAvP + policy gradients	8.77	9.20	**45.96**
ATSPrivacy (7-4-15)	8.45	8.89	38.61
ATSPrivacy (21-13-3,7-4-15)	**6.70**	**6.69**	36.42

- Computation and comm. cost.

Method	CIFAR-100 dataset Rounds(35%)	CIFAR-100 dataset Time(35%)
FedAvg + Default	300	1.05 hours
FedAvg + RandAugment	300	1.62 hours
FedAvg + TrivialAugment	450	2.17 hours
FedAvP (Fast Update)	200	1.18 hours
FedAvP	200	4.01 hours

Method	CIFAR-100 dataset Before(MB)	CIFAR-100 dataset Per round(MB)
FedAvg	0.00	15.35
FedMix	1.27	15.35
FedFA	0.00	15.38
FedAvP (Fast Update)	0.00	15.73
FedAvP	0.00	17.47

Thank you

Code: https://github.com/alsdml/FedAvP