PGFed: Personalize Each Client's Global Objective for Federated Learning

University of Pittsburgh
UNIVERSITY OF CENTRAL FLORIDA

Jun Luo†, Matias Mendieta‡, Chen Chen‡, and Shandong Wu†

†University of Pittsburgh, Pittsburgh, PA, USA ‡University of Central Florida, Orlando, FL, USA
(jul117@pitt.edu, wus3@upmc.edu) (matias.mendieta@ucf.edu, chen.chen@crcv.ucf.edu)

ICCV23 PARIS

Full paper GitHub

Introduction

- In existing personalized federated learning (FL) methods with **heterogeneous data**, *the way in which the **collaborative knowledge** transfers from the server to the clients is **implicit**.*

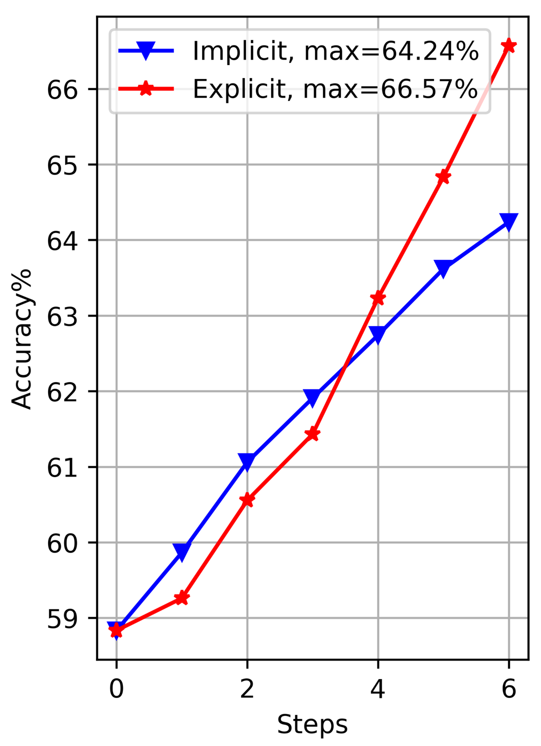
✓ **Collaborative knowledge**: non-local information
  - E.g., $F(\theta) = \sum p_i F_i(\theta)$

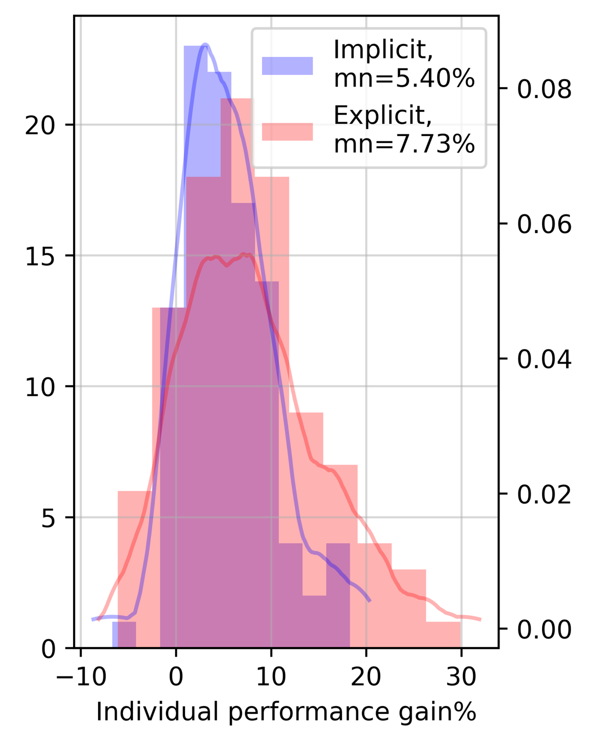
✓ **Explicitness** (as opposite of **implicitness**): Direct engagement with multiple clients' empirical risks. (explicit since not embed non-local info into model weights)
  - E.g., Global objective of FedAvg ($F_i(\theta) = f_i(\theta)$)
  - Update of personalized models (in pFL) can hardly be explicit (compute $f_j(\theta_i), \forall i,j \in [N]$ requires $O(N^2)$ communication overhead)

- Observation from experiments indicates benefits of ***explicit*** knowledge transfer

✓ Explicit (e.g.): $F_i(\theta_i) = f_i(\theta_i) + \frac{\mu}{N-1}\sum_{j\neq i} f_j(\theta_i)$

✓ Implicit: $F_i(\theta_i) = f_i(\theta_i)$ (local model of FedAvg)

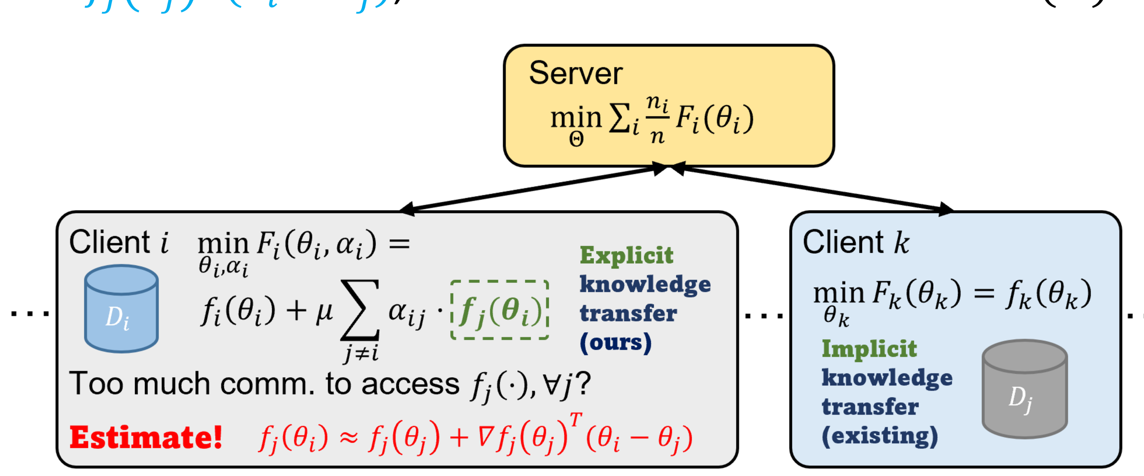




- Issues with the easy fix:

✓ Constant coefficients? Use adaptive coefficients $\alpha_{ij} \forall i,j \in [N]$

✓ $O(N^2)$ communication cost? Estimate $f_j(\theta_i) \approx f_j(\theta_j) + \nabla f_j(\theta_j)^T(\theta_i - \theta_j)$, reduces communication cost to $O(N)$



- Proposed **Personalized Global Federated Learning (PGFed)**

✓ Explicit knowledge transfer

✓ Adaptive coefficients

✓ $O(N)$ communication overhead

✓ Up to 15.47% accuracy boost and up to 4.2x convergence speedup over SOTA

Method

- **Objectives of Personalized Global Federated Learning (PGFed)**

✓ Global objective: $\min_{\Theta, A} F(\Theta, A) = \min_{\theta_1,\ldots,\theta_N,\alpha_1,\ldots,\alpha_N} \sum_{i=1}^{N} p_i F_i(\theta_i, \alpha_i)$

✓ Local objective: $F_i(\theta_i, \alpha_i) = f_i(\theta_i) + \mu \sum_{j\in[N]} \alpha_{ij} f_j(\theta_i)$

✓ Plugging $f_j(\theta_i) \approx f_j(\theta_j) + \nabla f_j(\theta_j)^T(\theta_i - \theta_j)$ into Local objective:

$$F_i(\theta_i, \alpha_i) \approx f_i(\theta_i) + \mathcal{R}_{aux}^{[N]}(\theta_i, \alpha_i)$$

$$\mathcal{R}_{aux}^{[N]}(\theta_i, \alpha_i) = \mu \sum_{j\in[N]} \alpha_{ij} \left(f_j(\theta_j) + \nabla_{\theta_j} f_j(\theta_j)^T(\theta_i - \theta_j)\right)$$

✓ Intuition behind why the approximation might work
  - Non-local risks restrain the personalized model weights from ungoverned drifting
  - More regularized updates of personalized models → approximation works

- **Gradient-based update**

✓ **W.r.t θ_i:**
$$\nabla_{\theta_i} F_i(\theta_i, \alpha_i) = \nabla_{\theta_i} f_i(\theta_i) + \nabla_{\theta_i}\mathcal{R}_{aux}^{[N]}(\theta_i, \alpha_i) = \nabla_{\theta_i} f_i(\theta_i) + \underbrace{\mu \sum_{j\in[N]} \alpha_{ij} \nabla_{\theta_j} f_j(\theta_j)}_{\tilde{g}_{[N]}}.$$

  - $\tilde{g}_{[N]}$ can be computed by the server with:
    - Client i uploading α_i
    - Client j uploading local gradient

✓ **W.r.t α_{ij}:**
$$\nabla_{\alpha_{ij}} F_i(\theta_i, \alpha_i) = \mu\left(f_j(\theta_j) + \nabla_{\theta_j} f_j(\theta_j)^T(\theta_i - \theta_j)\right) = \underbrace{\mu\left(f_j(\theta_j) - \nabla_{\theta_j} f_j(\theta_j)^T\theta_j\right)}_{g_\alpha^{(1)}} + \underbrace{\mu \nabla_{\theta_j} f_j(\theta_j)^T\theta_i}_{g_\alpha^{(2)}}.$$

  - $g_\alpha^{(1)}$ (scalar) can be computed and uploaded by client j
  - To compute the exact value of $g_\alpha^{(2)}$ needs to transmit all gradients to client i (takes $O(N^2)$ comm.)
    - Estimate: $g_\alpha^{(2)} \approx \tilde{g}_{[N]}^T \theta_i = \frac{\mu}{N}\left(\sum_{j\in[N]} \nabla_{\theta_j} f_j(\theta_j)\right)^T \theta_i$
    - Compute by server: save comm. and comp.
    - Compute locally: more accurate

- **To accommodate to M selected clients per round**

✓ $[N] \to S_t$ (selected set of clients in round t)

$$\tilde{g}_{S_t} = \mu \sum_{j\in S_t} \alpha_{ij} \nabla_{\theta_j} f_j(\theta_j) \qquad \tilde{g}_{S_t} = \frac{\mu}{M}\left(\sum_{j\in S_t} \nabla_{\theta_j} f_j(\theta_j)\right)$$

✓ Using momentum update to avoid losing previous rounds' info

$$\tilde{g}_{S_t}^i = (1-\beta)\tilde{g}_{S_t}^i(\text{downloaded}) + \beta \tilde{g}_{S_t}^i(\text{previous})$$

- **Detailed algorithm in full paper (QR code above)**

Experiments

- Mean top-1 local test accuracy on CIFAR10, CIFAR100, Dir(α=0.3), 25,50,100 clients

✓ **PGFed and PGFedMo boost the accuracy by up to 15.47%**

Algorithms	CIFAR10			CIFAR100		
	25 clients	50 clients	100 clients	25 clients	50 clients	100 clients
Local	72.40±0.45	70.28±0.38	67.39±0.20	32.74±0.08	26.05±0.34	23.06±0.47
FedAvg	65.07±0.25	64.41±0.66	63.19±0.46	28.48±0.59	26.06±0.65	25.58±0.80
FedDyn	67.31±0.36	65.02±0.91	62.49±0.06	34.17±0.43	27.06±0.18	23.88±0.36
pFedMe	70.60±0.23	68.92±0.35	66.40±0.04	27.97±0.24	23.82±0.06	22.35±0.03
FedFomo	72.33±0.03	72.17±0.48	70.86±0.27	32.15±0.61	25.90±1.17	24.48±0.44
APFL	77.03±0.26	77.36±0.18	76.29±0.13	39.16±0.93	35.15±0.65	33.86±0.60
FedRep	76.85±0.44	76.03±0.17	72.30±0.52	33.43±0.80	26.86±0.39	22.76±0.45
LG-FedAvg	72.83±0.28	70.44±0.31	67.55±0.09	33.65±0.19	27.13±0.37	24.82±0.28
FedPer	77.84±0.18	77.76±0.22	75.01±0.20	35.22±0.67	28.63±0.70	25.56±0.26
Per-FedAvg	75.49±0.74	76.27±0.50	75.41±0.35	32.89±0.43	32.24±0.75	32.59±0.21
FedRoD	79.73±0.68	79.61±0.22	77.76±0.32	39.55±0.58	33.87±2.42	31.49±0.19
FedBABU	78.92±0.36	79.35±0.84	76.34±0.22	32.71±0.23	29.66±0.64	27.72±0.11
PGFed	**81.02±0.41**	**81.42±0.31**	**78.56±0.35**	**43.12±0.03**	**38.45±0.44**	**35.71±0.54**
PGFedMo	81.20±0.08	81.48±0.32	78.74±0.22	43.44±0.14	38.50±0.45	35.76±0.65

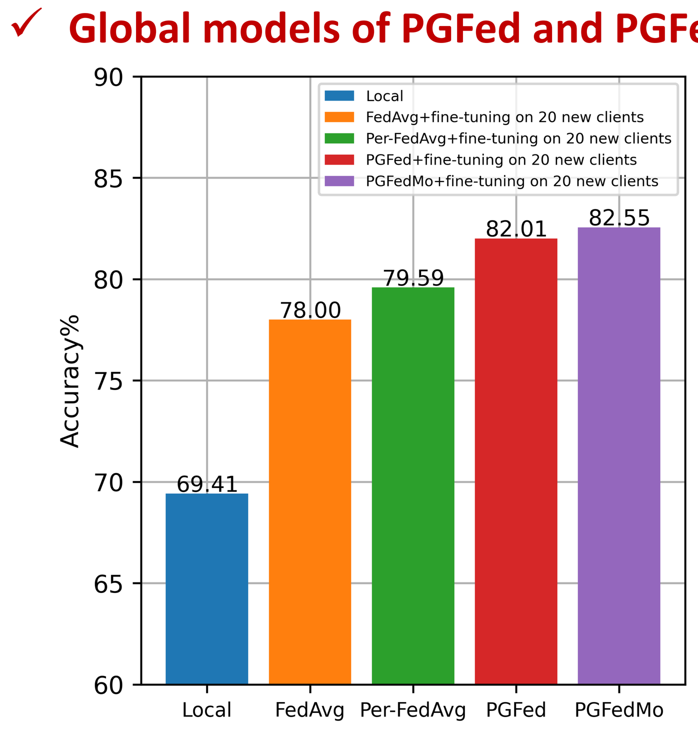
- Convergence speed (#round to reach 70% accuracy) and client individual gain

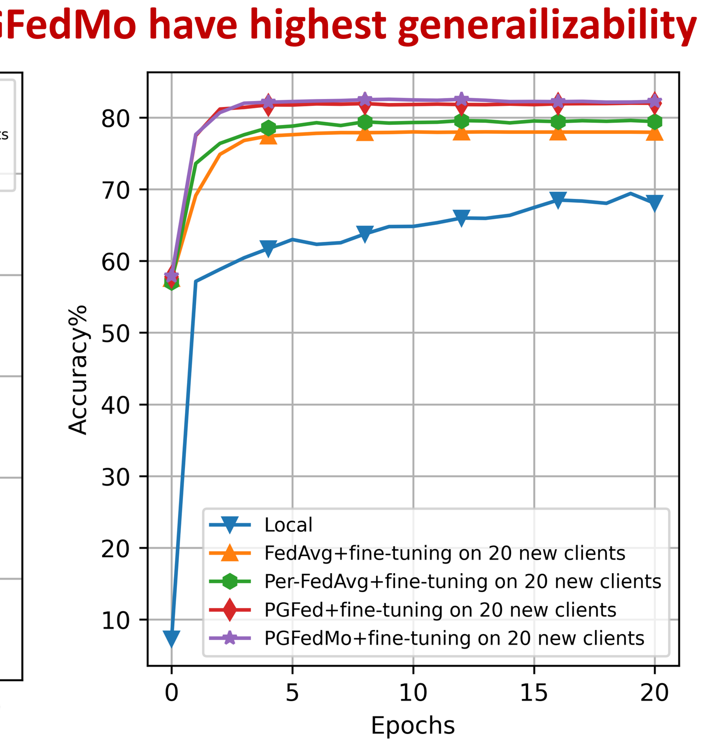
✓ **PGFed and PGFedMo have 3.7× average speedup with highest individual gain**

	25 clients			50 clients			100 clients		
	round	speed up	Individual gain	round	speed up	Individual gain	round	speed up	Individual gain
Fedavg	∞	N/A	−8.99±10.36	∞	N/A	−8.90±15.48	∞	N/A	−5.02±14.30
APFL	31	1.0×	2.79±8.07	28	1.7×	5.73±8.43	24	2.6×	8.37±6.91
FedPer	8	3.9×	5.31±2.56	6	7.8×	8.31±6.00	8	7.9×	8.63±5.26
Per-FedAvg	31	1.0×	0.72±6.22	47	1.0×	5.02±7.39	63	1.0×	8.09±7.00
FedRoD	26	1.2×	7.80±3.68	35	1.3×	8.84±6.29	10	6.3×	10.68±6.14
PGFed	9	3.4×	8.49±4.67	14	3.4×	10.78±5.88	15	4.2×	11.15±5.06
PGFedMo	9	3.4×	8.61±3.59	14	3.4×	10.90±6.11	15	4.2×	11.16±5.44

- Fine-tuning on 20 new clients the output global model from SOTA pFL algorithms

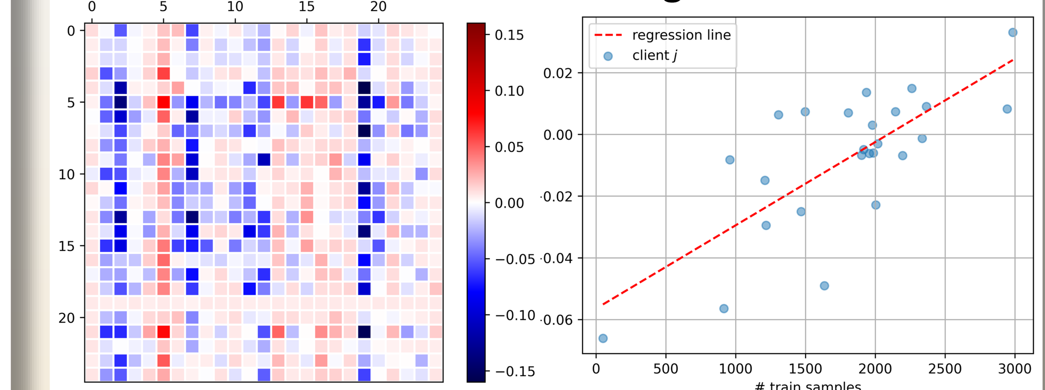
✓ **Global models of PGFed and PGFedMo have highest generalizability**





Experiments (cont'd)

- Visualization of coefficients and their relationship with local training set sizes



- Mean top-1 local test accuracy on OrganAMNIST

	25 clients sample 50% Dir(1.0)	50 clients sample 25% Dir(0.3)	100 clients sample 25% Dir(0.3)
Local	90.45±0.19	90.63±0.07	87.14±0.10
FedAvg	99.11±0.03	98.74±0.04	98.47±0.08
APFL	97.49±0.05	97.53±0.06	96.19±0.11
FedRep	95.06±0.16	94.86±0.07	92.47±0.04
LGFedAvg	90.47±0.18	90.99±0.08	87.52±0.22
FedPer	97.89±0.06	97.55±0.08	95.56±0.33
Per-FedAvg	98.40±0.02	96.80±0.04	95.09±0.07
FedRoD	98.61±0.05	98.14±0.09	97.05±0.06
FedBABU	96.49±0.28	94.33±0.13	91.07±0.23
PGFed	99.20±0.04	99.17±0.05	98.94±0.02
PGFedMo	99.21±0.04	99.17±0.07	98.86±0.06

- Communication- & computation-efficient PGFed

	Images/s	Relative speed	Accuracy
FedAvg	6917.1	100.00%	64.41±0.66
APFL	3389.8	48.99%	77.36±0.18
Per-FedAvg	3464.5	50.09%	76.27±0.50
FedRoD	6682.4	96.61%	79.61±0.22
PGFed	6120.0	88.48%	81.42±0.31
PGFedMo	6032.8	87.22%	81.48±0.32
PGFed-CE	6175.5	89.28%	81.16±0.56

- More experiments in full paper (QR code above)

Conclusion

- We observed that explicit knowledge transfer generalize better than its implicit counterpart
- Proposed explicit PGFed and PGFedMo achieve high performance with $O(N)$ comm.
- Future studies include further reducing comm. for personalized FL

Acknowledgements

- NIH/NIC 1R01CA218405
- NSF CICI: SIVD: 2115082
- NSF/NIH 1R01EB032896
- NIH 3R01EB032896-03S1
- XSEDE by NSF ACI-1548562
- Bridges-2 by NSF ACI-1928147
- NSF/Intel Partnership on MLWiNS 2003198