

Large Model-Assisted Collaborative Learning for Vertical Domain Small Models

- **Name:** Jianqing Zhang
- **Age:** 26
- **Ph.D.:** Shanghai Jiao Tong University
- **M.S.:** Shanghai Jiao Tong University
- **Collaborations:**
 - Tsinghua University
 - Queen's University Belfast
 - Louisiana State University

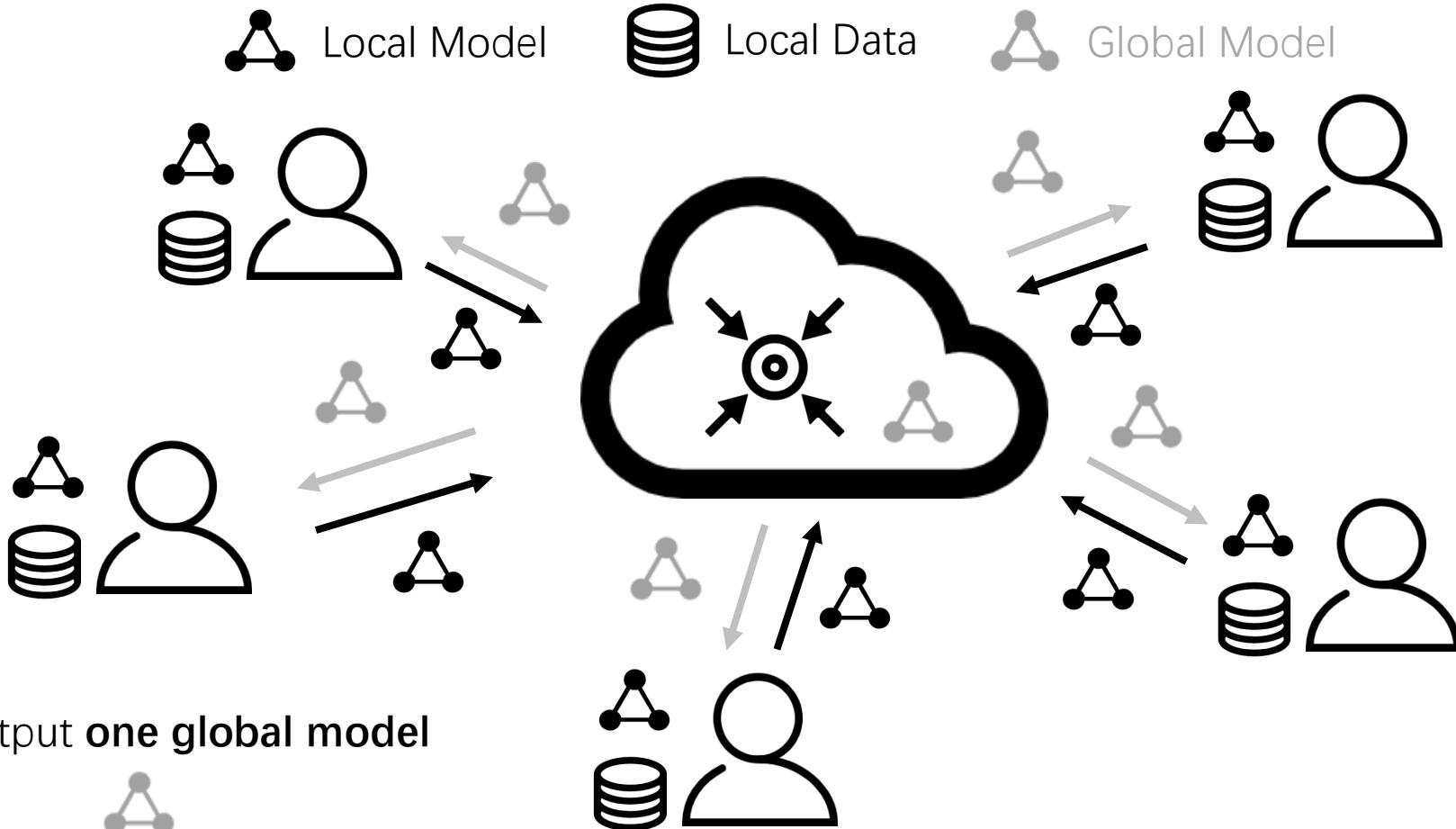


Content

- **Research interests**
 - Federated Learning (Collaboration of Vertical Domain Small Models)
 - Privacy-Preserving Synthetic Dataset Generation (Large Model)
 - Large Model-Assisted Collaborative Learning for Vertical Domain Small Models
- **Projects**
 - PFLlib (1300+ stars, 200+ forks), HtFLLib, FL-IoT, etc.
- **Featured publications (first author)**
 - Stage ① [Personalized Federated Learning]:
 - PFLlib, AAAI'23, KDD'23, ICCV'23, NeurIPS'23
 - Stage ② [Heterogeneous Federated Learning]:
 - HtFLLib, AAAI'24, submitted to NeurIPS'24
 - Stage ③ [Privacy-Preserving Synthetic Dataset Generation]:
 - submitted to EMNLP'24
 - Stage ④ [Generative Model-Assisted Collaborative Learning]:
 - CVPR'24

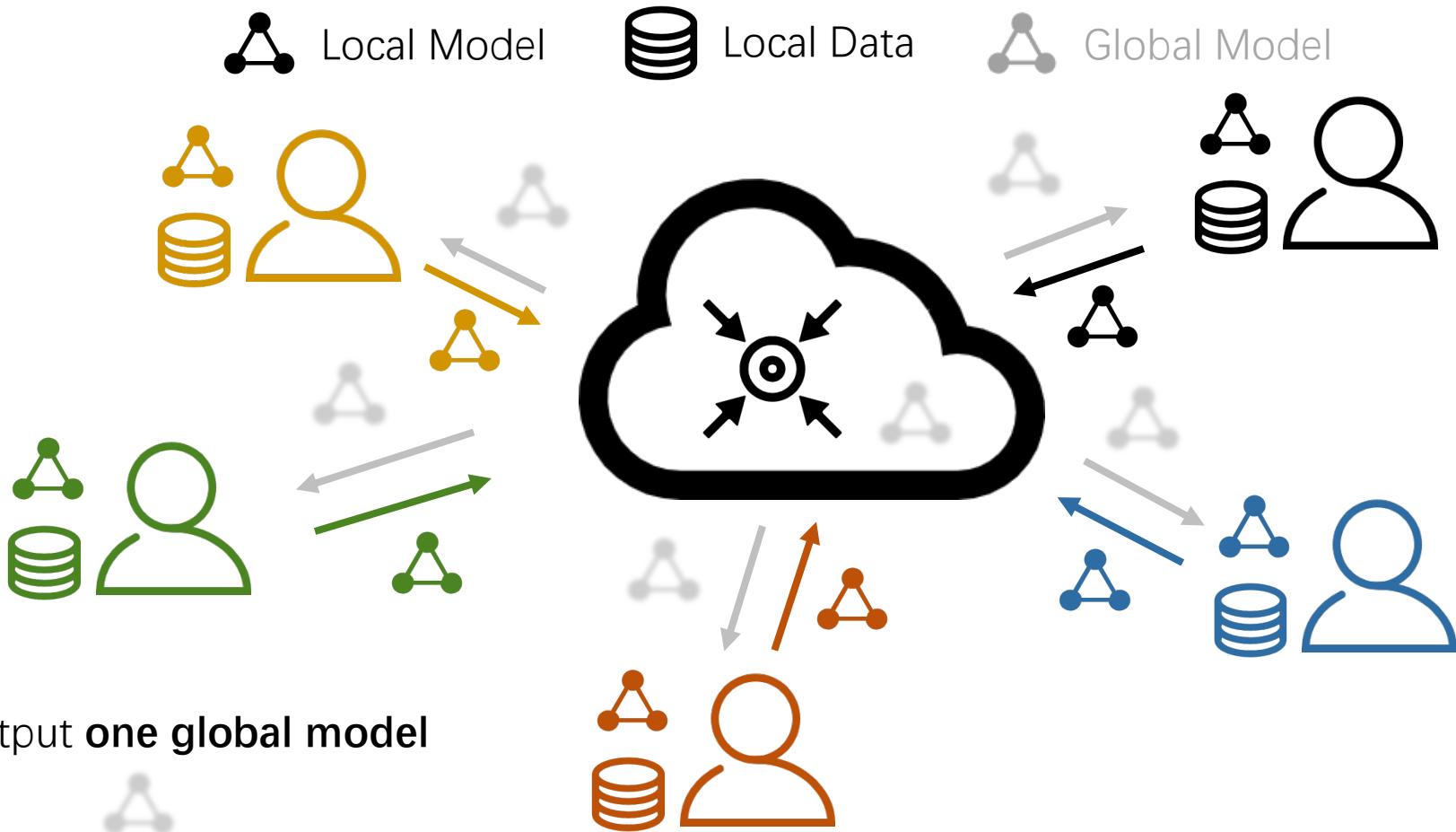
Federated Learning (FL)

- A **collaborative-learning** and **privacy-preserving** technique
- Learn an AI model among clients by **only sharing models** with the server.



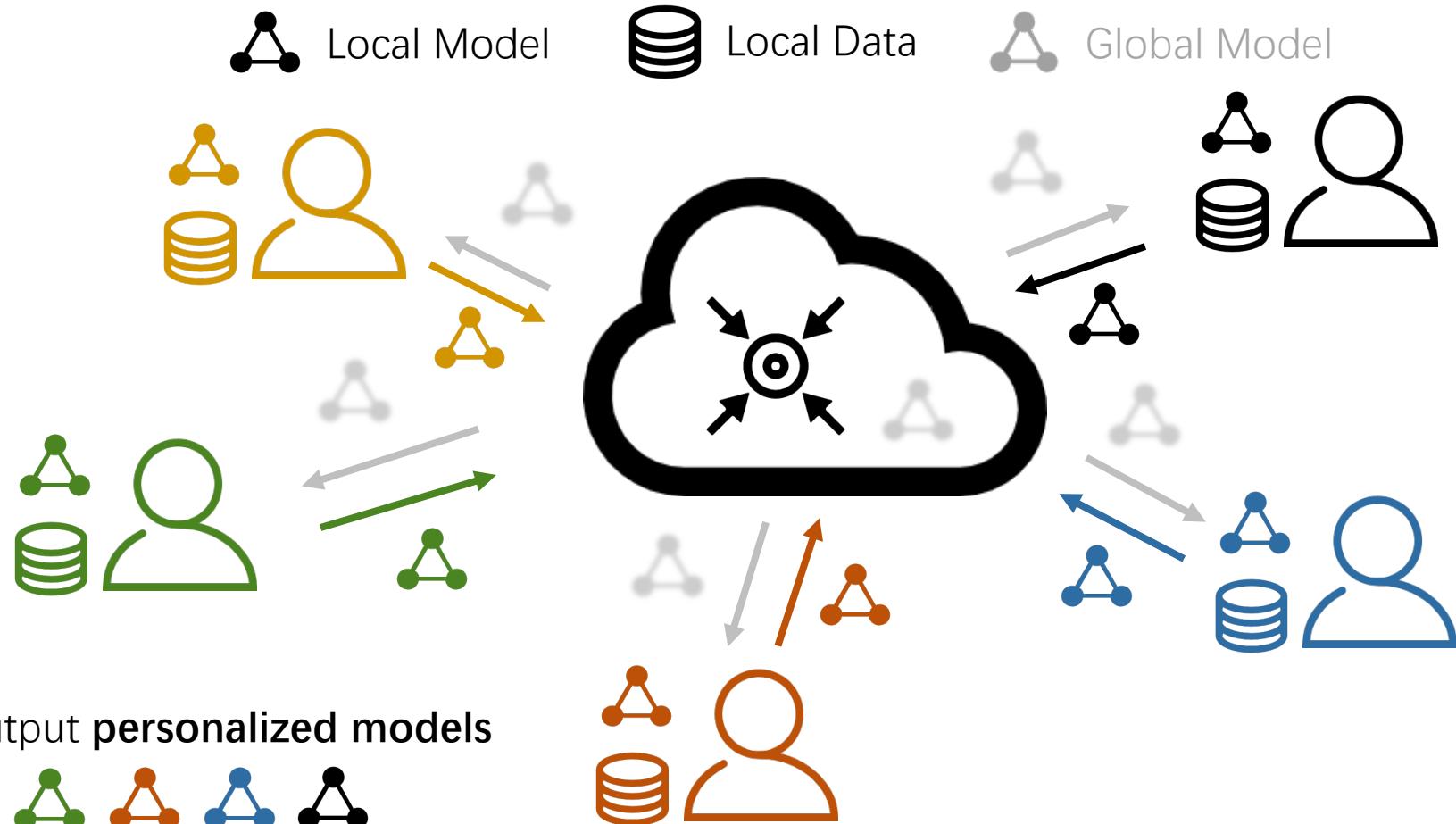
① Data heterogeneity in FL

- Data is **generated by different clients** and forbidden to be manipulated
- Each client has **personalized preferences**



① Data heterogeneity in FL

- Data is **generated by different clients** and forbidden to be manipulated
- Each client has **personalized preferences**



① PFLlib: personalized FL (pFL) algorithm library

- Beginner-friendly
- Comprehensive (37 FLs&pFLs)
- Popular (1300+ stars)
- ...

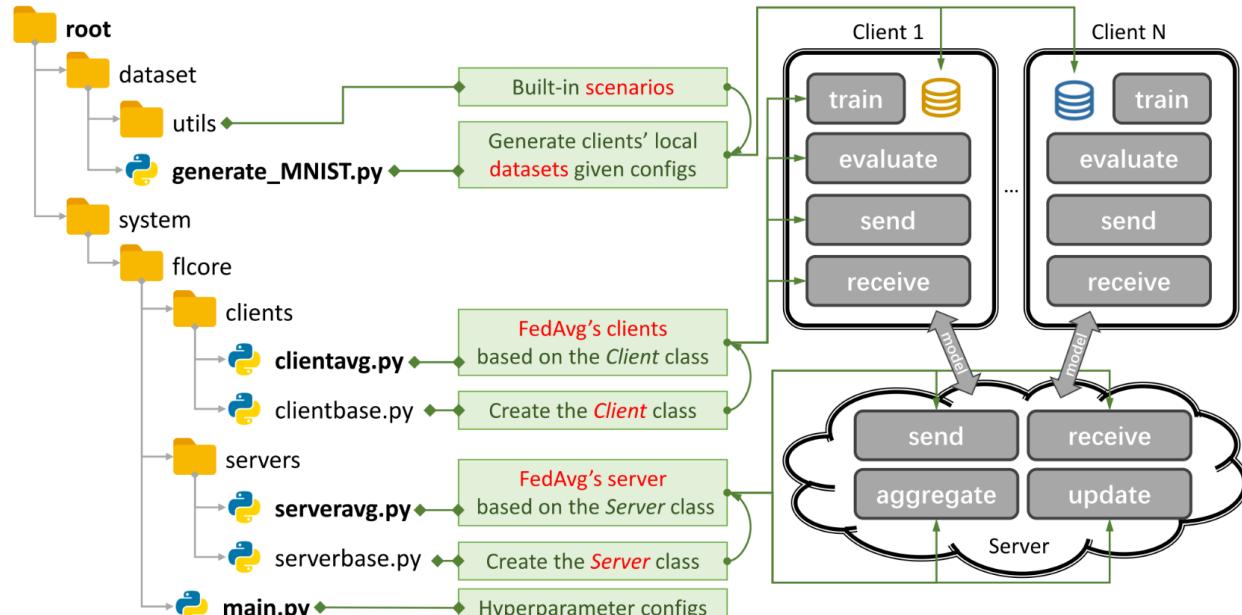
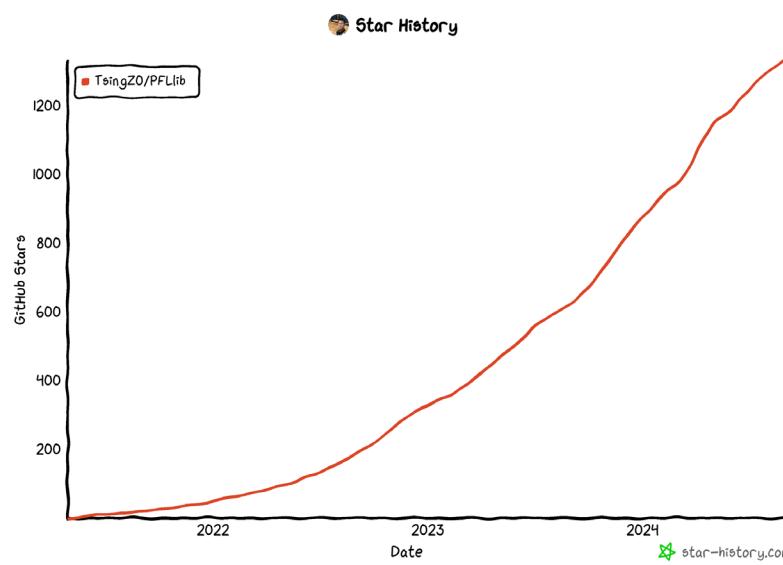


Figure 1: An Example for FedAvg. You can create a scenario using `generate_DATA.py` and run an algorithm using `main.py`, `clientNAME.py`, and `serverNAME.py`.

We've created a user-friendly algorithm library and evaluation platform for those new to federated learning. Join us in expanding the FL community by contributing your algorithms, datasets, and metrics to this project.

- 37 traditional FL ([tFL](#)) or personalized FL ([pFL](#)) algorithms, 3 scenarios, and 20 datasets.
- Some experimental results are available [here](#).
- Refer to [this guide](#) to learn how to use it.
- This library can simulate scenarios using the 4-layer CNN on Cifar100 for 500 clients on one NVIDIA GeForce RTX 3090 GPU card with only 5.08GB GPU memory cost.

① Publications

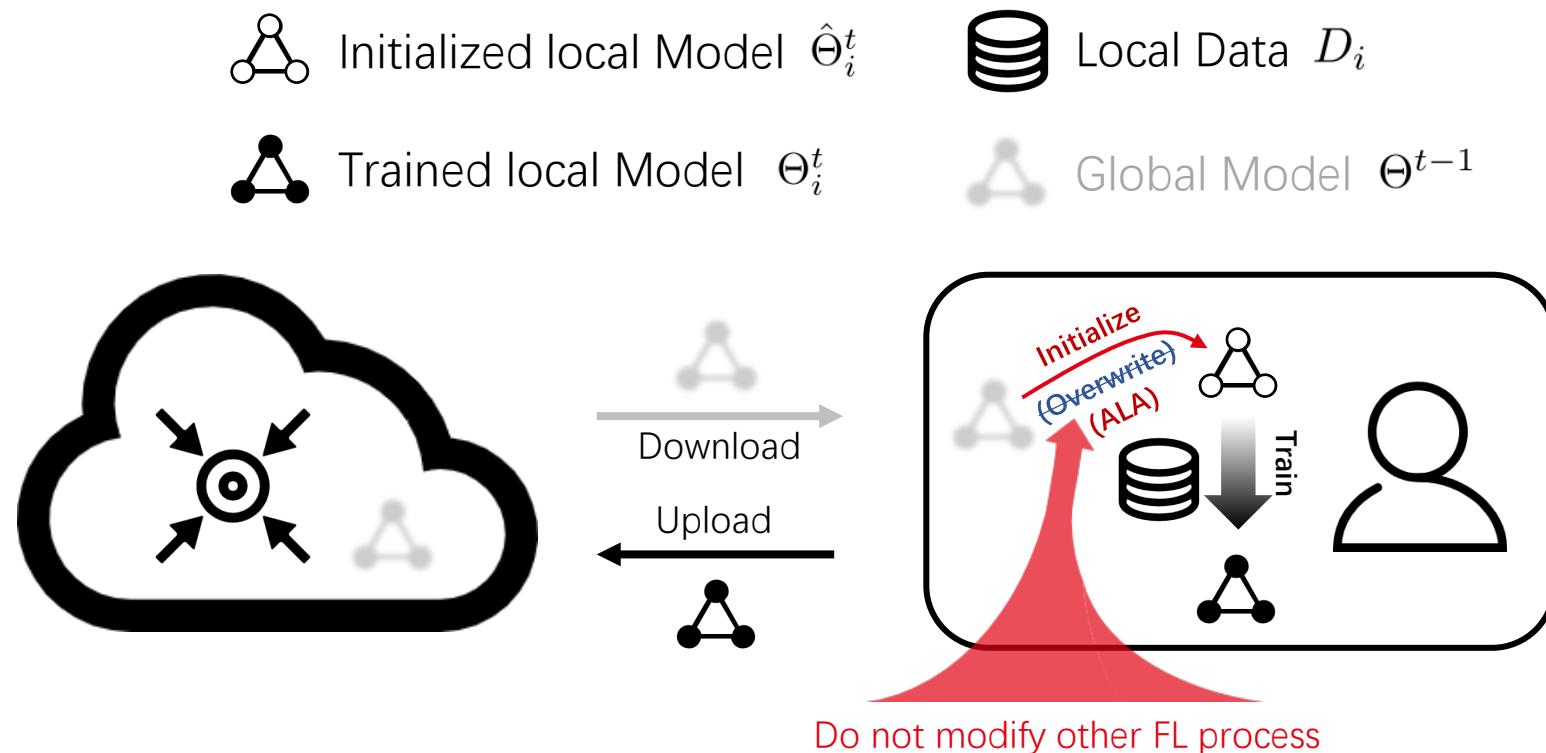
- [AAAI'23] FedALA: Adaptive Local Aggregation for Personalized Federated Learning.
- [KDD'23] FedCP: Separating Feature Information for Personalized Federated Learning via Conditional Policy.
- [ICCV'23] GPFL: Simultaneously Learning Generic and Personalized Feature Information for Personalized Federated Learning.
- [NeurIPS'23] Eliminating Domain Bias for Federated Learning in Representation Space.
- How can we consider both generalization and personalization?

① Publications

- [AAAI'23] FedALA: Adaptive Local Aggregation for Personalized Federated Learning.
- [KDD'23] FedCP: Separating Feature Information for Personalized Federated Learning via Conditional Policy.
- [ICCV'23] GPFL: Simultaneously Learning Generic and Personalized Feature Information for Personalized Federated Learning.
- [NeurIPS'23] Eliminating Domain Bias for Federated Learning in Representation Space.

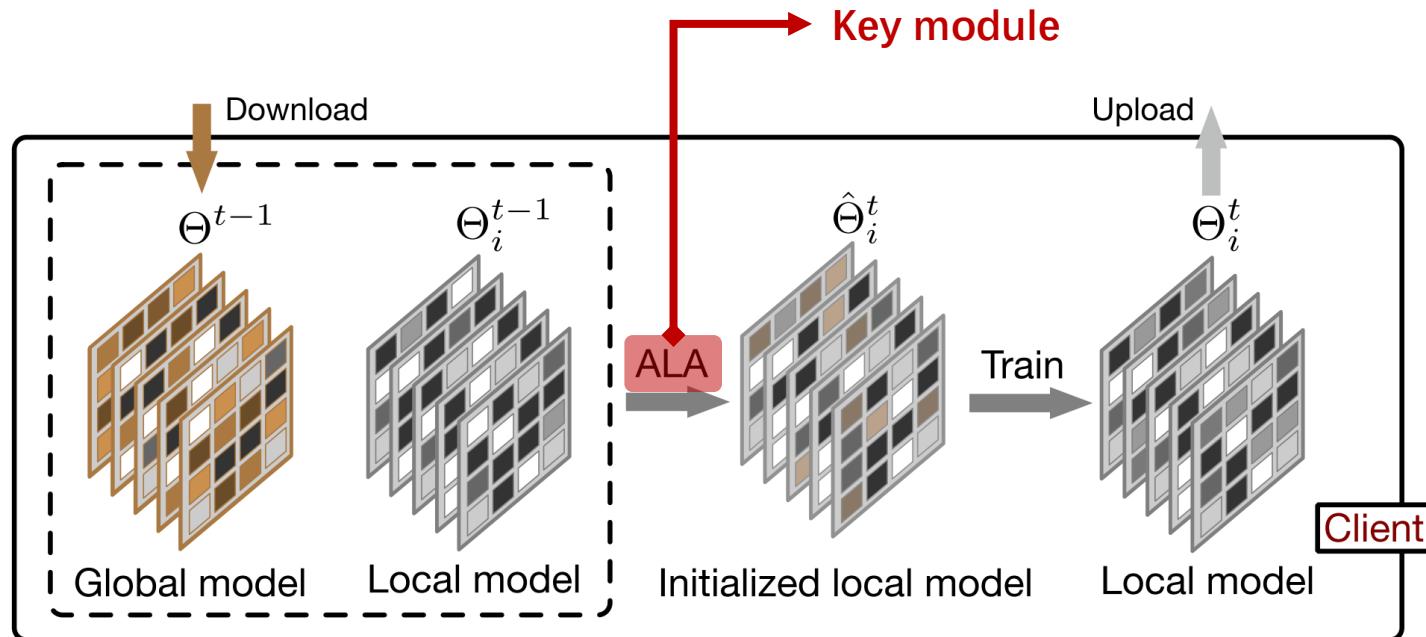
Motivation of FedALA

- Original workflow in FL
 - Both the **desired** and **undesired** information exist in the global model, resulting in **poor generalization ability**



FedALA

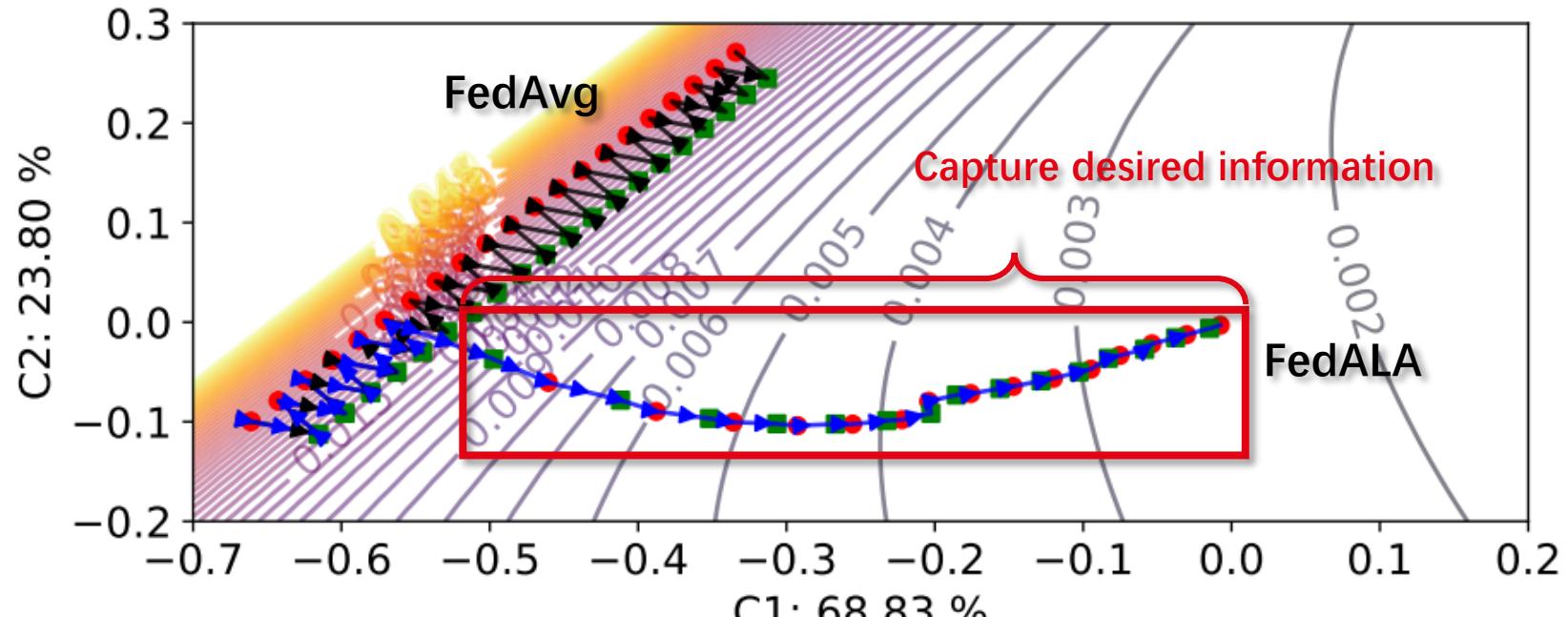
- Extract each client's desired information from the global model that facilitates local training
- Adaptively aggregate the information in the global and local model for initialization



Workflow on the client in one iteration

FedALA

- Learning trajectory on one client: **FedAvg** vs. **FedALA**
- Activate ALA in the subsequent iterations



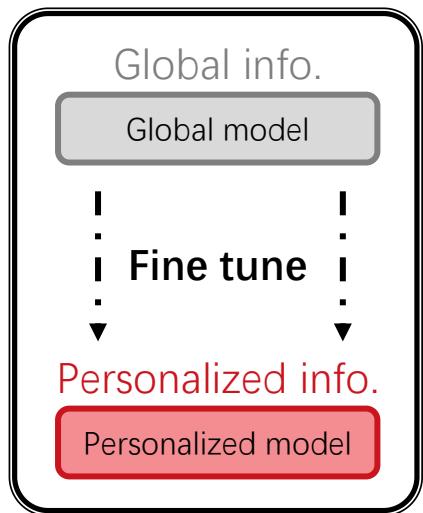
2D visualization of local learning trajectory

① Publications

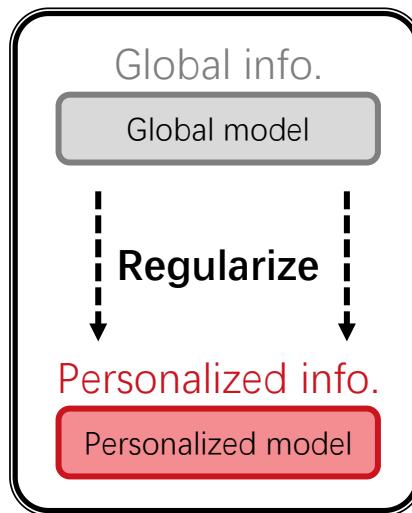
- [AAAI'23] FedALA: Adaptive Local Aggregation for Personalized Federated Learning.
- [KDD'23] FedCP: Separating Feature Information for Personalized Federated Learning via Conditional Policy.
- [ICCV'23] GPFL: Simultaneously Learning Generic and Personalized Feature Information for Personalized Federated Learning.
- [NeurIPS'23] Eliminating Domain Bias for Federated Learning in Representation Space.

Existing pFL

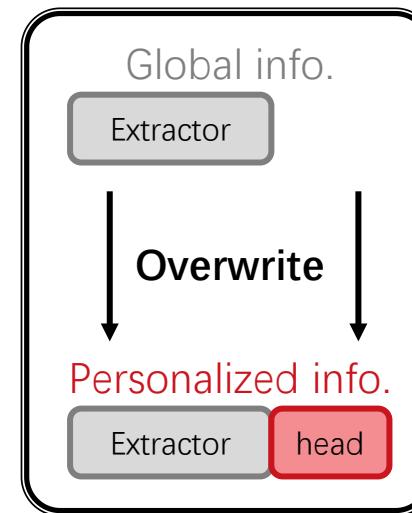
- **Consensus:** reasonably utilizing global and personalized information is the key for pFL.
 - meta-learning-based (Per-FedAvg), regularization-based (Ditto), and personalized-head-based (FedRep) pFL.



Per-FedAvg[1]



Ditto[2]



FedRep[3]

- They only focus on model parameters, but **ignore the source of information: data.**

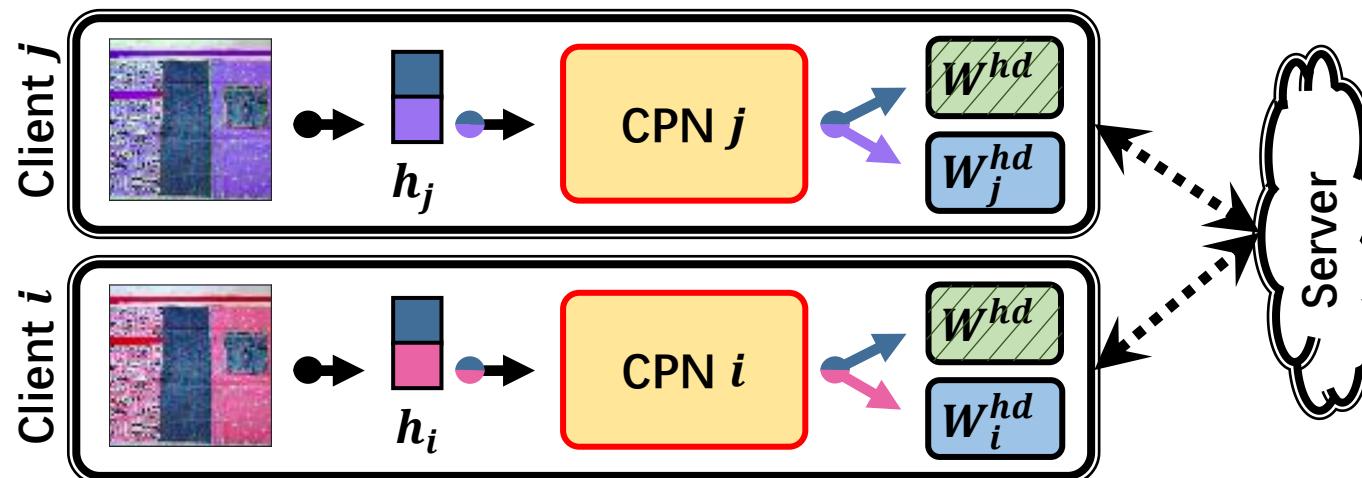
[1] Fallah A, Mokhtari A, Ozdaglar A. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. NeurIPS, 2020.

[2] Li T, Hu S, Beirami A, et al. Ditto: Fair and robust federated learning through personalization. ICML, 2021.

[3] Collins L, Hassani H, Mokhtari A, et al. utilizing shared representations for personalized federated learning. ICML, 2021.

FedCP

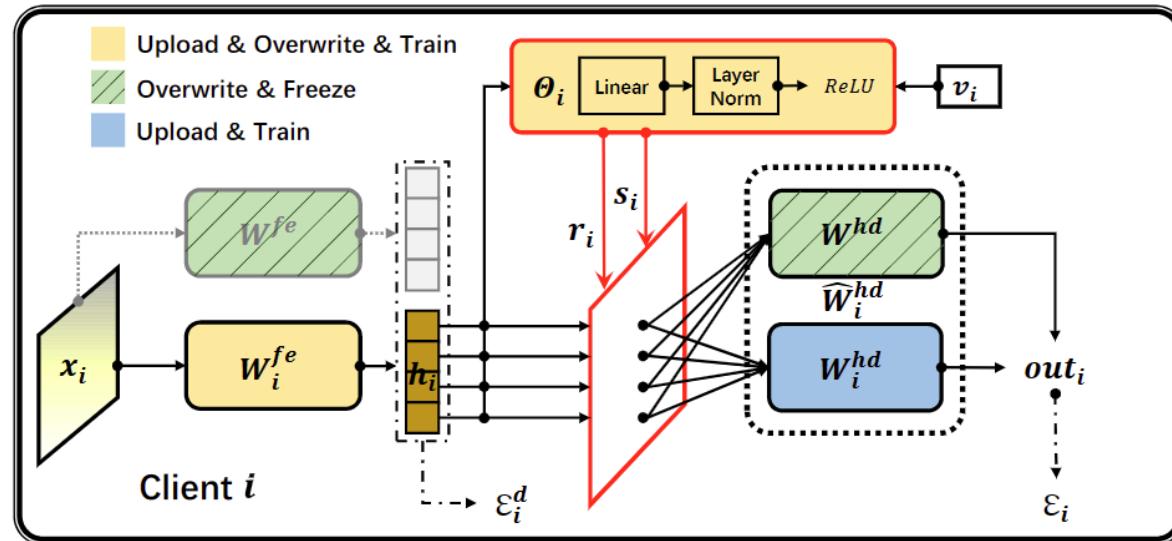
- We separate feature information via an *auxiliary Conditional Policy Network (CPN)*.
 - Generate *sample-specific policy*
 - *End-to-end training* together with the client model
 - *Lightweight* (e.g., 4.67% parameters of ResNet-18)



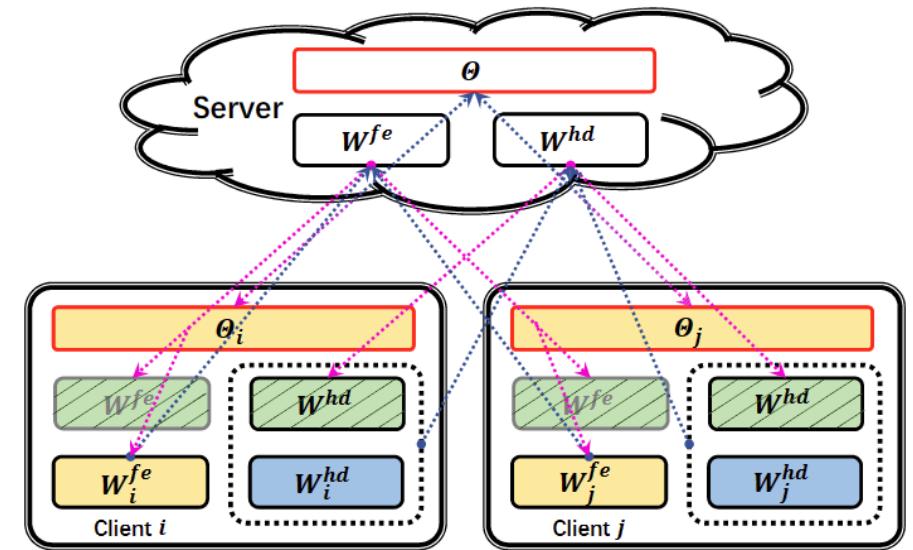
- We utilize *global and personalized information* via global and personalized heads.

FedCP

- Architecture



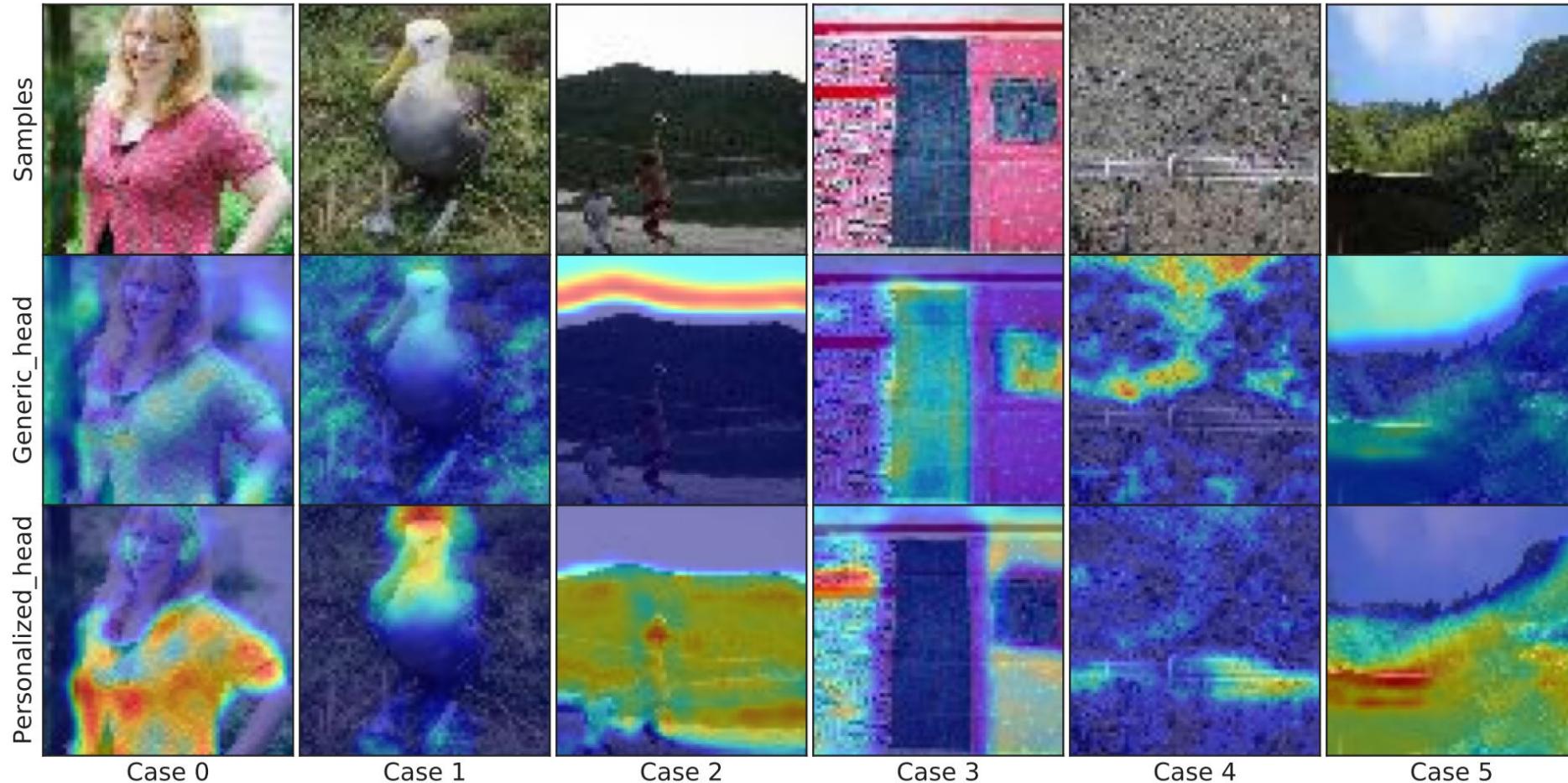
Data flow in the personalized model



Upload and download stream

FedCP

- Separating Feature Information



Six samples from the Tiny-ImageNet dataset

① Publications

- [AAAI'23] FedALA: Adaptive Local Aggregation for Personalized Federated Learning.
- [KDD'23] FedCP: Separating Feature Information for Personalized Federated Learning via Conditional Policy.
- [ICCV'23] GPFL: Simultaneously Learning Generic and Personalized Feature Information for Personalized Federated Learning.
- [NeurIPS'23] Eliminating Domain Bias for Federated Learning in Representation Space.

GPFL

- GPFL **introduces more global information** during local training to enhance local model
- CoV **eliminates the interaction between** global and personalized feature learning

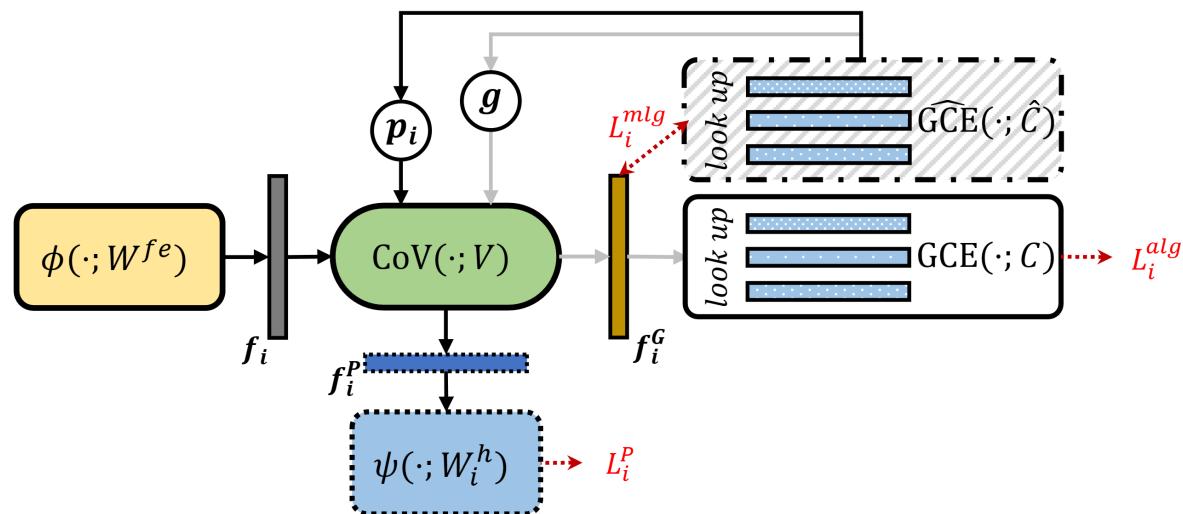
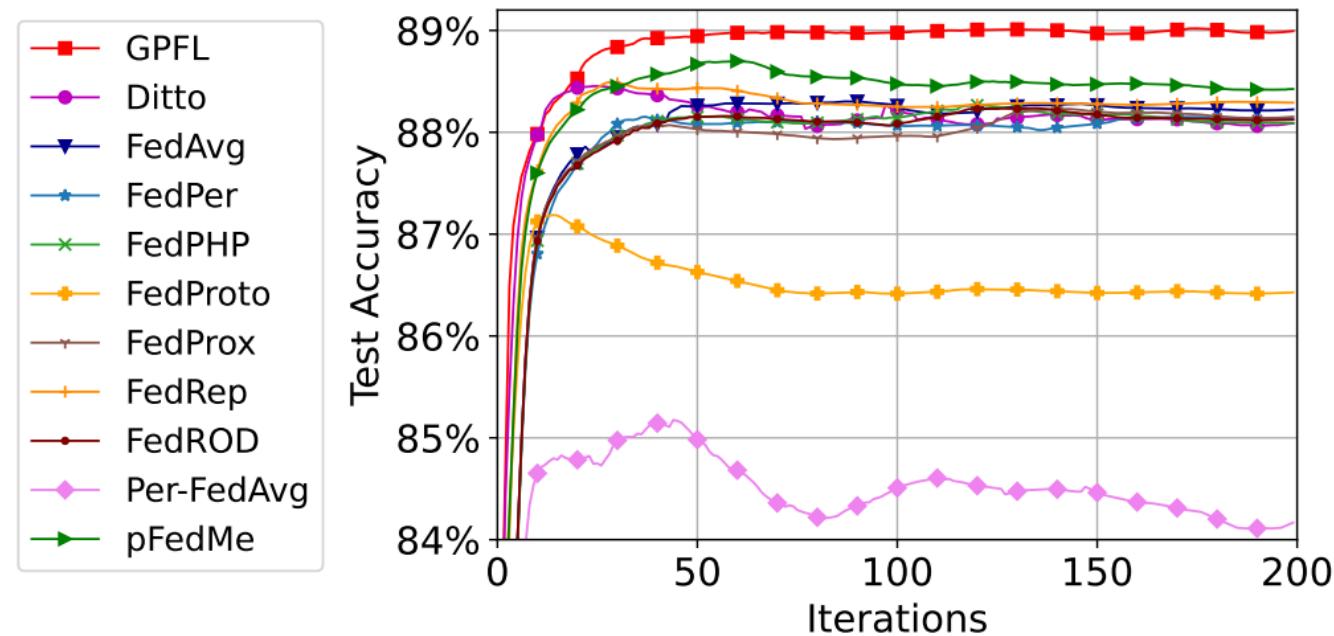


Illustration of client modules and data flow between them

GPFL

- Address the **overfitting** issue in pFL



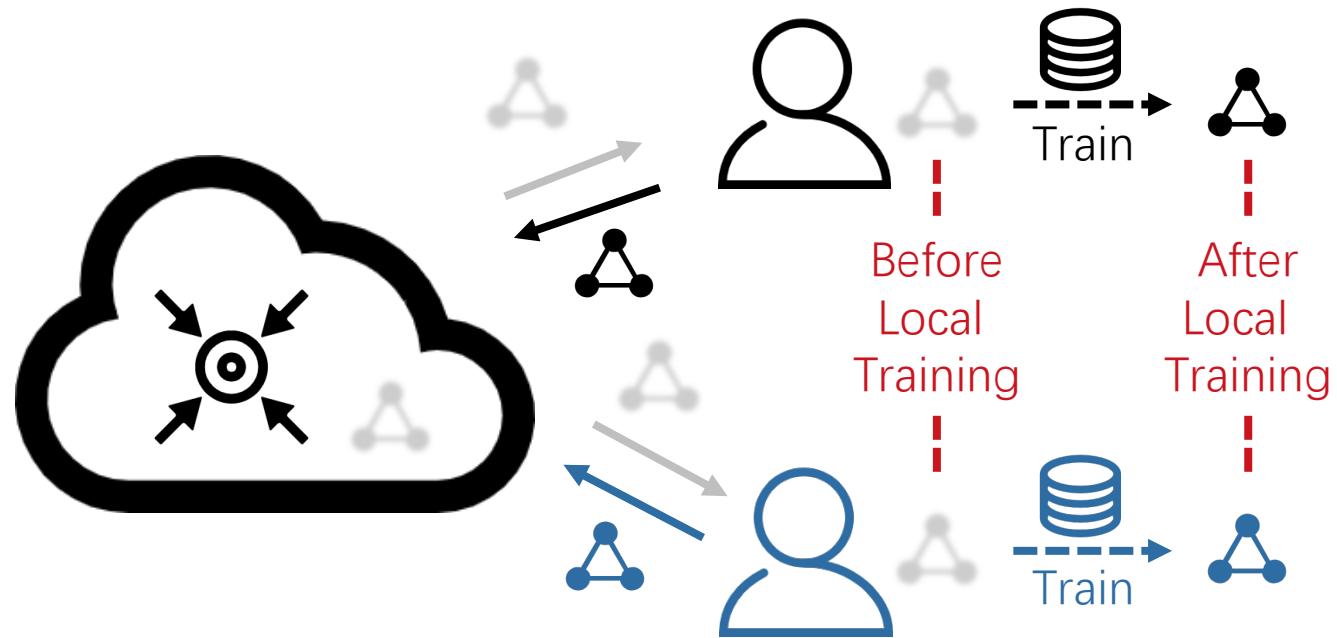
Test accuracy curves in the feature shift setting

① Publications

- [AAAI'23] FedALA: Adaptive Local Aggregation for Personalized Federated Learning.
- [KDD'23] FedCP: Separating Feature Information for Personalized Federated Learning via Conditional Policy.
- [ICCV'23] GPFL: Simultaneously Learning Generic and Personalized Feature Information for Personalized Federated Learning.
- [NeurIPS'23] Eliminating Domain Bias for Federated Learning in Representation Space.

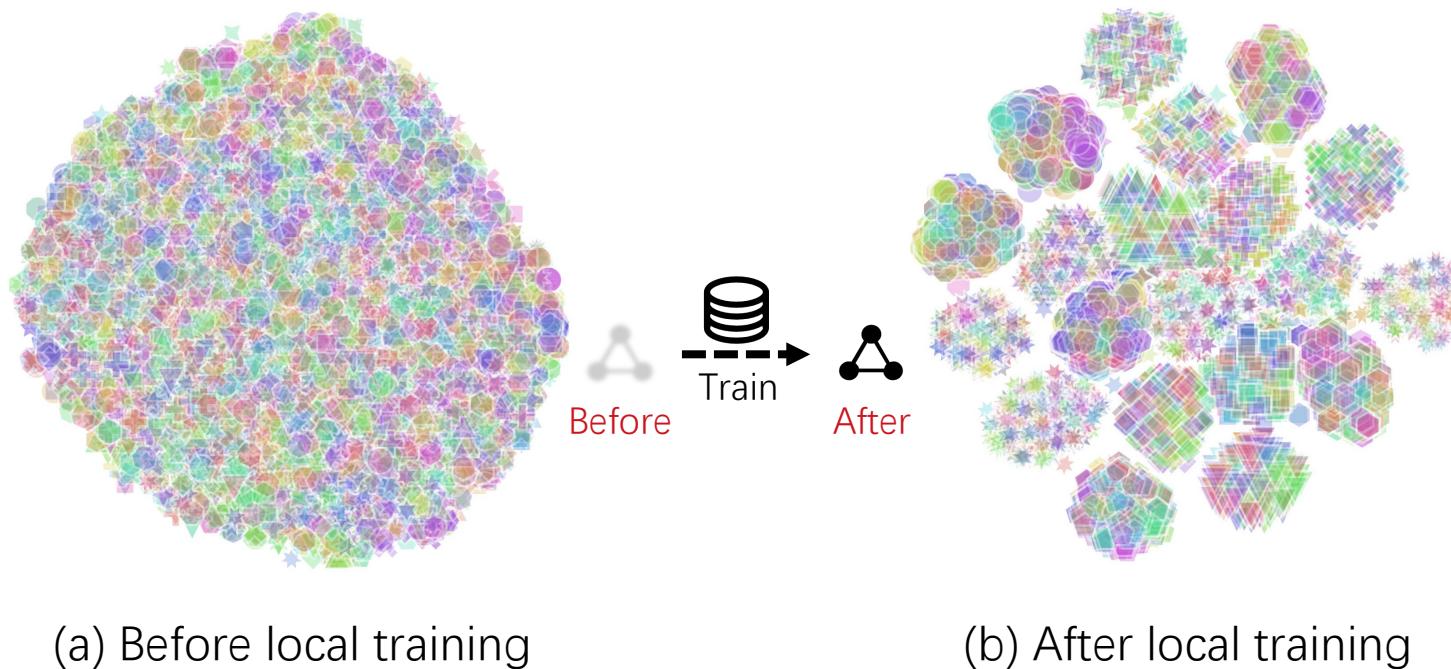
Local training

- Clients' local training turns the received global model to client-specific local models



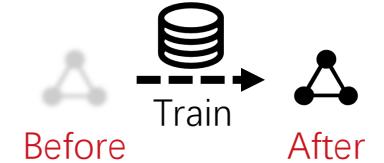
Representation bias phenomenon

- After local training, the feature representations are **biased** to client-specific domains

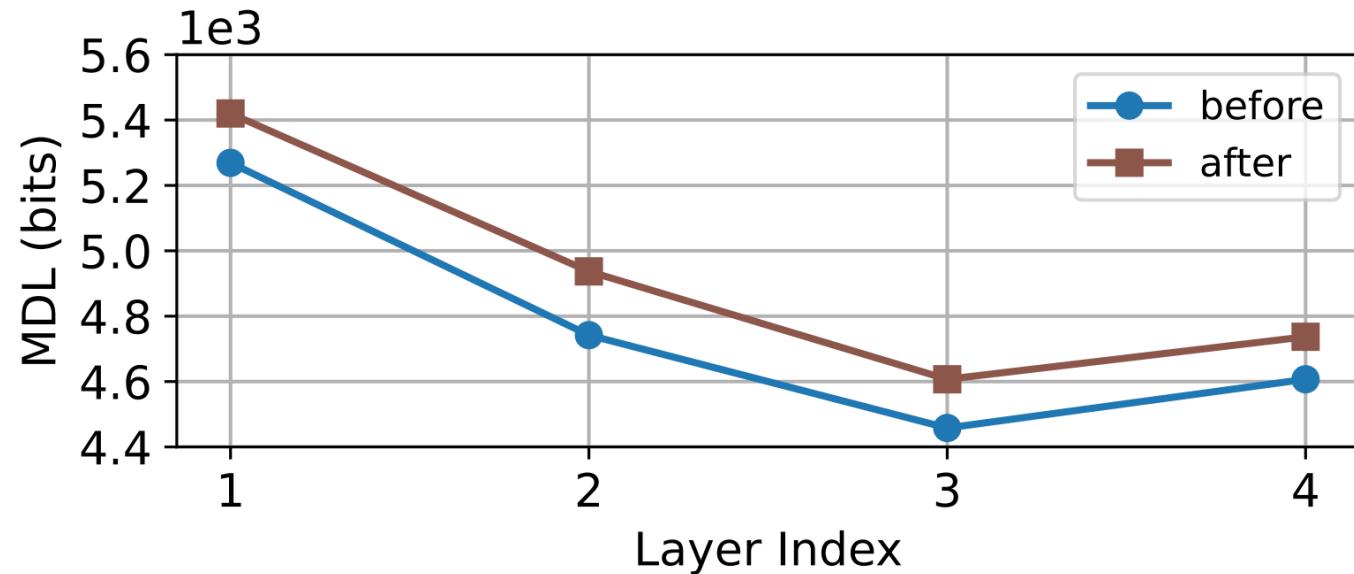


We use *color* and *shape* to distinguish *labels* and *clients*, respectively.

Representation degeneration phenomenon



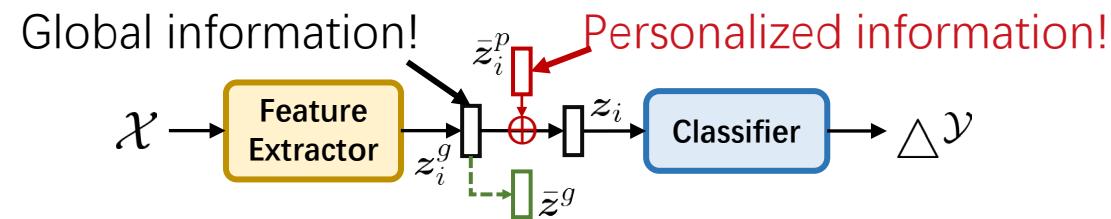
- At the same time, representations' quality is also **degenerated**



Per-layer MDL (bits) for representations before/after local training in FedAvg.
A large MDL value means low representation quality.

DBE

- Eliminate **domain bias**
- Improve **bi-directional knowledge transfer**



Local model (with PRBM and MR)

DBE

- **Local-to-global** knowledge transfer

Corollary 1. Consider a local data domain \mathcal{D}_i and a virtual global data domain \mathcal{D} for client i and the server, respectively. Let $\mathcal{D}_i = \langle \mathcal{U}_i, c^* \rangle$ and $\mathcal{D} = \langle \mathcal{U}, c^* \rangle$, where $c^* : \mathcal{X} \mapsto \mathcal{Y}$ is a ground-truth labeling function. Let \mathcal{H} be a hypothesis space of VC dimension d and $h : \mathcal{Z} \mapsto \mathcal{Y}, \forall h \in \mathcal{H}$. When using DBE, given a feature extraction function $\mathcal{F}^g : \mathcal{X} \mapsto \mathcal{Z}$ that shared between \mathcal{D}_i and \mathcal{D} , a random labeled sample of size m generated by applying \mathcal{F}^g to a random sample from \mathcal{U}_i labeled according to c^* , then for every $h^g \in \mathcal{H}$, with probability at least $1 - \delta$:

$$\mathcal{L}_{\mathcal{D}}(h^g) \leq \mathcal{L}_{\hat{\mathcal{D}}_i}(h^g) + \sqrt{\frac{4}{m} \left(d \log \frac{2em}{d} + \log \frac{4}{\delta} \right)} + d_{\mathcal{H}}(\tilde{\mathcal{U}}_i^g, \tilde{\mathcal{U}}^g) + \lambda_i,$$

where $\mathcal{L}_{\hat{\mathcal{D}}_i}$ is the empirical loss on \mathcal{D}_i , e is the base of the natural logarithm, and $d_{\mathcal{H}}(\cdot, \cdot)$ is the \mathcal{H} -divergence between two distributions. $\lambda_i := \min_{h^g} \mathcal{L}_{\mathcal{D}}(h^g) + \mathcal{L}_{\mathcal{D}_i}(h^g)$, $\tilde{\mathcal{U}}_i^g \subseteq \mathcal{Z}$, $\tilde{\mathcal{U}}^g \subseteq \mathcal{Z}$, and $d_{\mathcal{H}}(\tilde{\mathcal{U}}_i^g, \tilde{\mathcal{U}}^g) \leq d_{\mathcal{H}}(\tilde{\mathcal{U}}_i, \tilde{\mathcal{U}})$. $\tilde{\mathcal{U}}_i^g$ and $\tilde{\mathcal{U}}^g$ are the induced distributions of \mathcal{U}_i and \mathcal{U} under \mathcal{F}^g , respectively. $\tilde{\mathcal{U}}_i$ and $\tilde{\mathcal{U}}$ are the induced distributions of \mathcal{U}_i and \mathcal{U} under \mathcal{F} , respectively. \mathcal{F} is the feature extraction function in the original FedAvg without DBE.

DBE

- **Global-to-local** knowledge transfer

Corollary 2. Let \mathcal{D}_i , \mathcal{D} , \mathcal{F}^g , and λ_i defined as in Corollary I. Given a translation transformation function $PRBM : \mathcal{Z} \mapsto \mathcal{Z}$ that shared between \mathcal{D}_i and virtual \mathcal{D} , a random labeled sample of size m generated by applying \mathcal{F}' to a random sample from \mathcal{U}_i labeled according to c^* , $\mathcal{F}' = PRBM \circ \mathcal{F}^g : \mathcal{X} \mapsto \mathcal{Z}$, then for every $h' \in \mathcal{H}$, with probability at least $1 - \delta$:

$$\mathcal{L}_{\mathcal{D}_i}(h') \leq \mathcal{L}_{\hat{\mathcal{D}}}(h') + \sqrt{\frac{4}{m} \left(d \log \frac{2em}{d} + \log \frac{4}{\delta} \right)} + d_{\mathcal{H}}(\tilde{\mathcal{U}}', \tilde{\mathcal{U}}'_i) + \lambda_i,$$

where $d_{\mathcal{H}}(\tilde{\mathcal{U}}', \tilde{\mathcal{U}}'_i) = d_{\mathcal{H}}(\tilde{\mathcal{U}}^g, \tilde{\mathcal{U}}_i^g) \leq d_{\mathcal{H}}(\tilde{\mathcal{U}}, \tilde{\mathcal{U}}_i) = d_{\mathcal{H}}(\tilde{\mathcal{U}}_i, \tilde{\mathcal{U}})$. $\tilde{\mathcal{U}}'$ and $\tilde{\mathcal{U}}'_i$ are the induced distributions of \mathcal{U} and \mathcal{U}_i under \mathcal{F}' , respectively.

Please refer to our paper for proofs.

② Model heterogeneity in FL

- Communication overhead, device heterogeneity, and **intellectual property**
- Transmits **lightweight knowledge carriers** instead of exposing model parameters



② HtFLlib: heterogeneous FL algorithm library

- Burgeoning
- Beginner-friendly
- Data-free
- ...

Scenarios and datasets

Here, we only show the MNIST dataset in the *label skew* scenario generated via Dirichlet distribution for example. Please refer to my other repository [PFLlib](#) for more help.

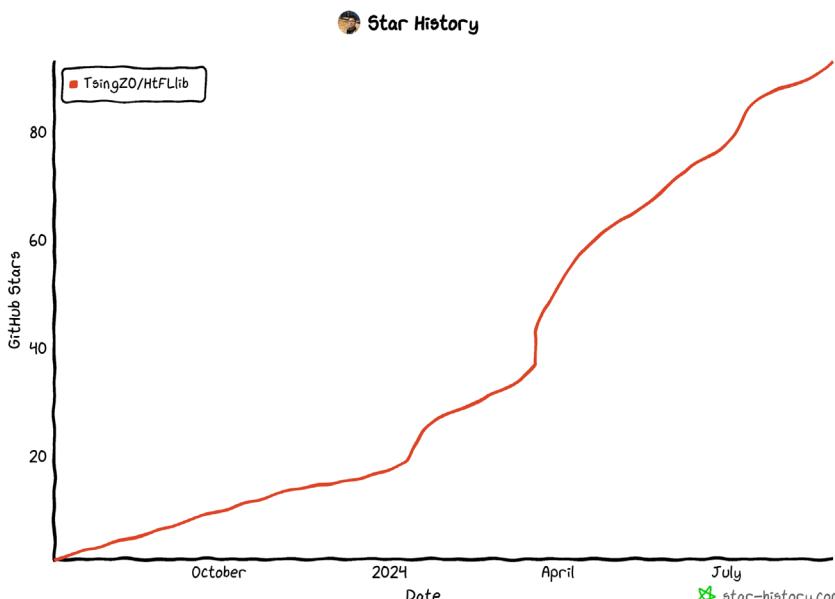
You can also modify codes in PFLlib to support model heterogeneity scenarios, but it requires much effort. In this repository, you only need to configure `system/main.py` to support model heterogeneity scenarios.

Note: you may need to manually clean checkpoint files in the `temp/` folder via `system/clean_temp_files.py` if your program crashes accidentally. You can also set a checkpoint folder by yourself to prevent automatic deletion using the `-sfn` argument in the command line.

Data-free algorithms with code (updating)

Here, "data-free" refers to the absence of any additional dataset beyond the clients' private data. We only consider data-free algorithms here, as they have fewer restrictions and assumptions, making them more valuable and easily extendable to other scenarios, such as the existence of public server data.

- Local — Each client trains its model locally without federation.
- FedDistill (FD) — [Communication-Efficient On-Device Machine Learning: Federated Distillation and Augmentation under Non-IID Private Data 2018](#)
- FML — [Federated Mutual Learning 2020](#)
- LG-FedAvg — [Think Locally, Act Globally: Federated Learning with Local and Global Representations 2020](#)
- FedGen — [Data-Free Knowledge Distillation for Heterogeneous Federated Learning ICML 2021](#)
- FedProto — [FedProto: Federated Prototype Learning across Heterogeneous Clients AAAI 2022](#)
- FedKD — [Communication-efficient federated learning via knowledge distillation Nature Communications 2022](#)
- FedGH — [FedGH: Heterogeneous Federated Learning with Generalized Global Header ACM MM 2023](#)
- FedTGP — [FedTGP: Trainable Global Prototypes with Adaptive-Margin-Enhanced Contrastive Learning for Data and Model Heterogeneity in Federated Learning AAAI 2024](#)
- FedKTL — [An Upload-Efficient Scheme for Transferring Knowledge From a Server-Side Pre-trained Generator to Clients in Heterogeneous Federated Learning CVPR 2024](#) (Note: FedKTL requires pre-trained generators to run, please refer to its [project page](#) for download links.)



② Publications

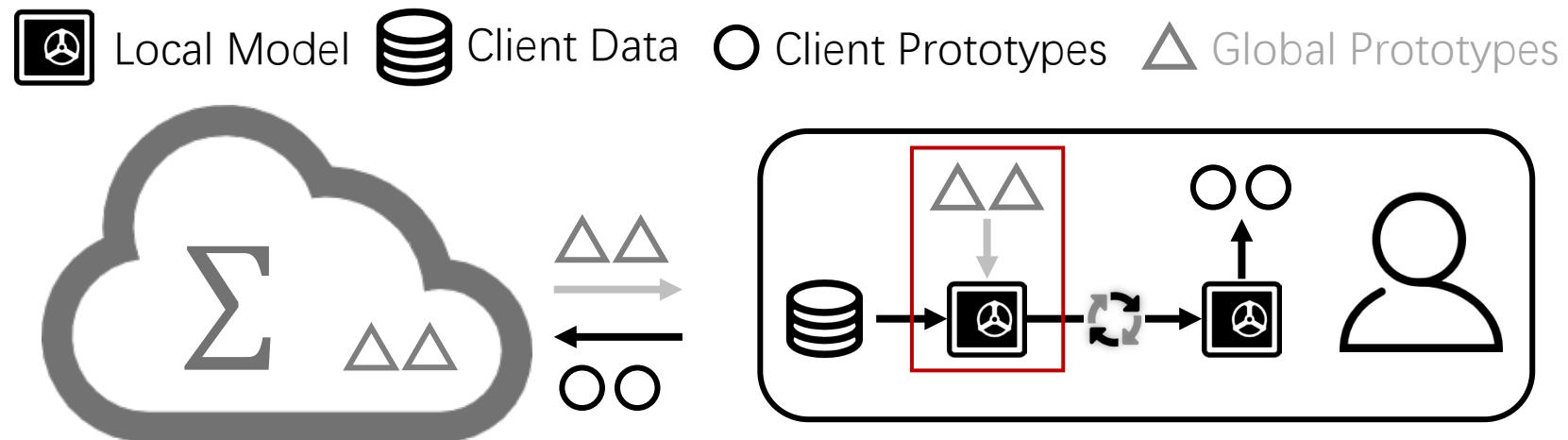
- [AAAI'24] FedTGP: Trainable Global Prototypes with Adaptive-Margin-Enhanced Contrastive Learning for Data and Model Heterogeneity in Federated Learning.
- [CVPR'24] An Upload-Efficient Scheme for Transferring Knowledge From a Server-Side Pre-trained Generator to Clients in Heterogeneous Federated Learning.
- How can knowledge be shared and aggregated to benefit participants?

② Publications

- [AAAI'24] FedTGP: Trainable Global Prototypes with Adaptive-Margin-Enhanced Contrastive Learning for Data and Model Heterogeneity in Federated Learning.
- How can knowledge be shared and aggregated to benefit participants?

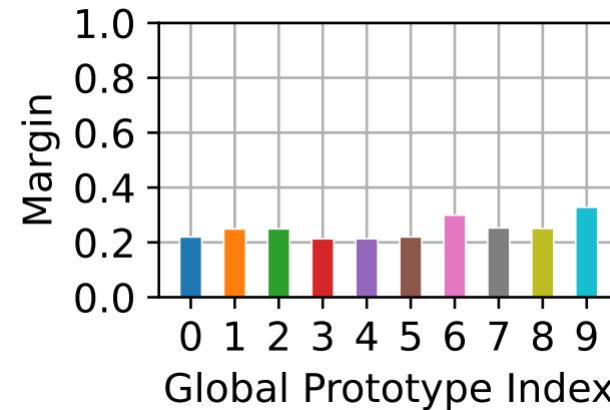
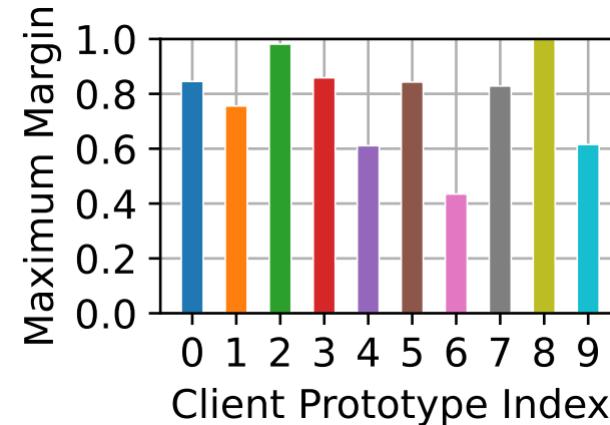
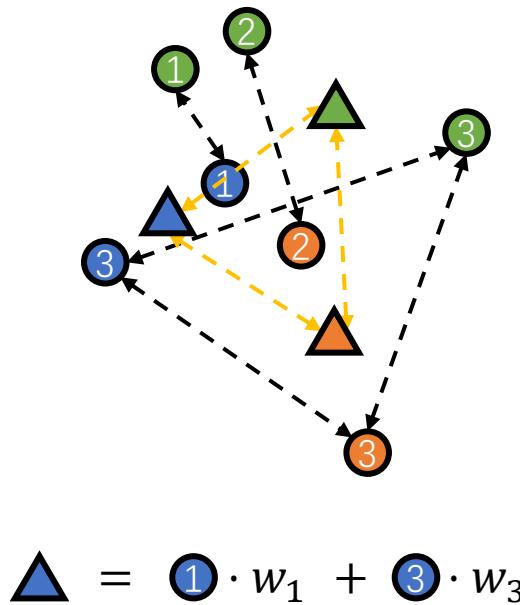
FedProto: share prototypes (class representatives)

- Share **client prototypes** with the server
- Aggregate client prototypes to generate **global prototypes**
- Train client models with both client data and global prototypes



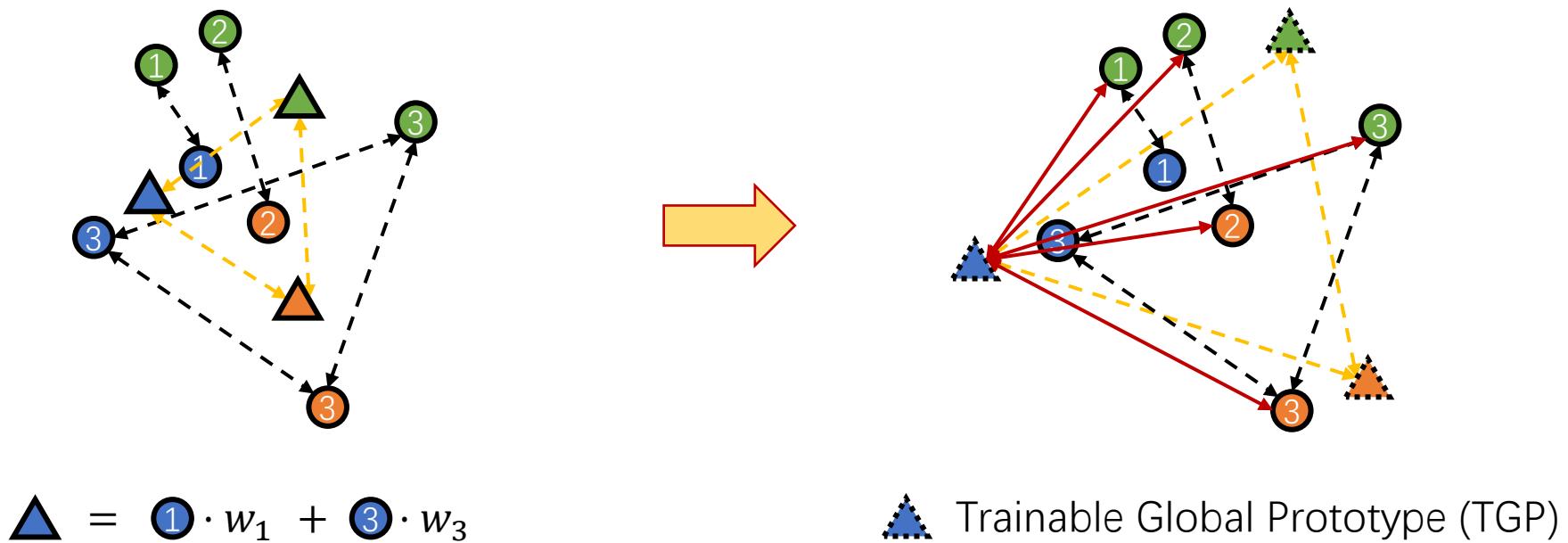
Issues of FedProto

- Global prototype (Δ) margin **shrinks** after weighted-averaging



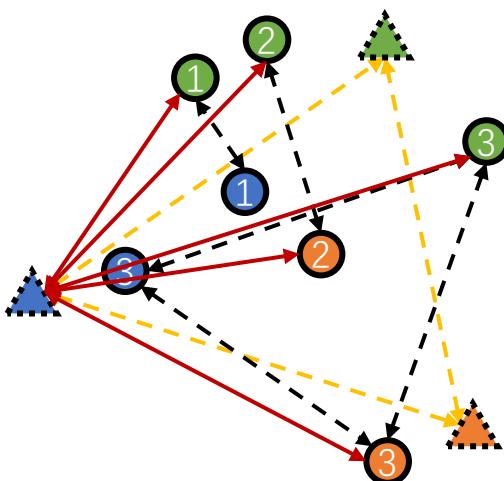
FedTGP

- Remove weighted-averaging
- Consider the uploaded client prototypes as data
- **Enlarge** the global prototype margin

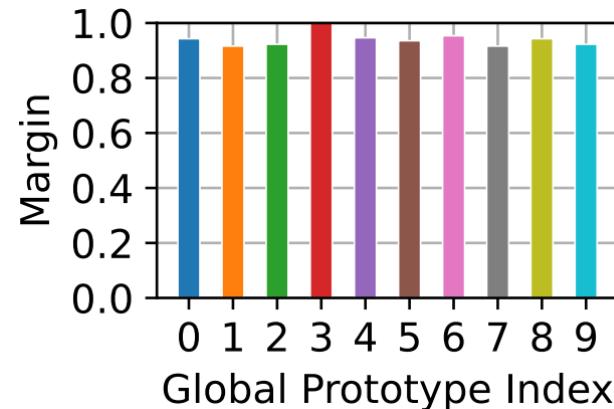
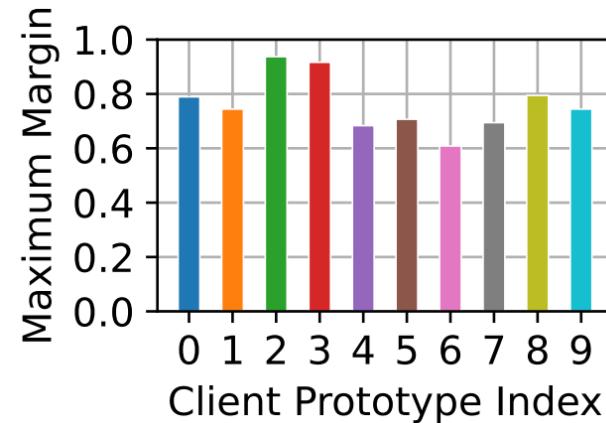


FedTGP

- Remove weighted-averaging
- Consider the uploaded client prototypes as data
- **Enlarge** the global prototype margin



▲ Trainable Global Prototype (TGP)



FedTGP

- Server objective: **Enlarge** the global prototype **margin** to improve discrimination
- **Train global prototypes** using **Adaptive-margin-enhanced Contrastive Learning (ACL)**

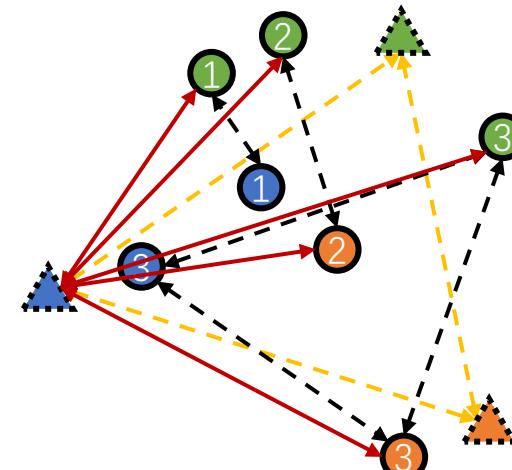
$$\min_{\hat{\mathcal{P}}} \sum_{c=1}^C \mathcal{L}_P^c,$$

$$\mathcal{L}_P^c = \sum_{i \in \mathcal{I}^t} -\log \frac{e^{-(\phi(P_i^c, \hat{P}^c) + \delta(t))}}{e^{-(\phi(P_i^c, \hat{P}^c) + \delta(t))} + \sum_{c'} e^{-\phi(P_i^c, \hat{P}^{c'})}}$$
$$\delta(t) = \min(\max_{c \in [C], c' \in [C], c \neq c'} \phi(Q_t^c, Q_t^{c'}), \tau),$$

$$Q_t^c = \frac{1}{|\mathcal{P}_t^c|} \sum_{i \in \mathcal{I}^t} P_i^c, \forall c \in [C]$$

τ is a margin threshold

maximum cluster margin



- ▲ \hat{P}^c : A TGP of class c
- ▲ $\hat{\mathcal{P}}$: All TGP
- P_i^c : A prototype of class c from client i

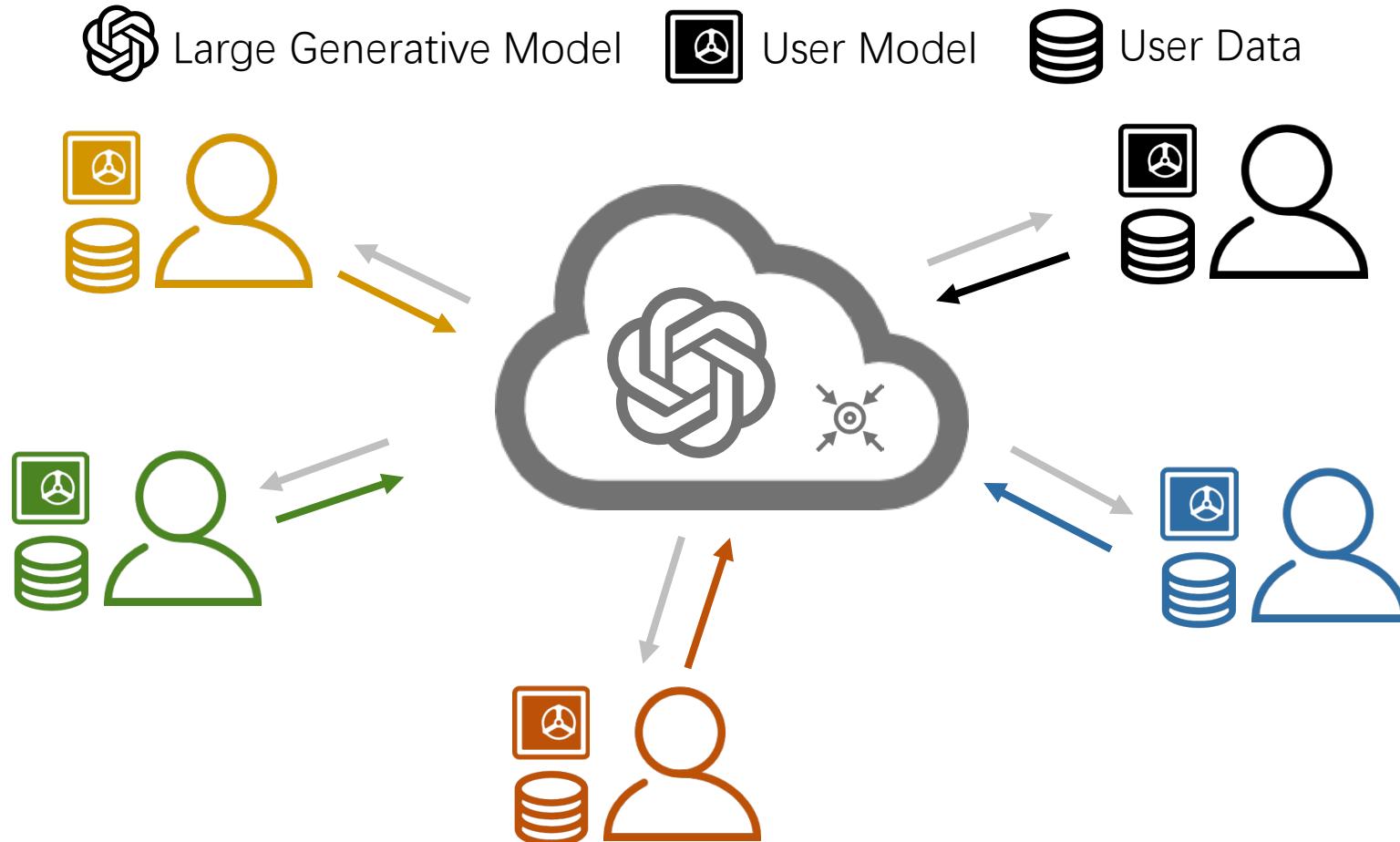
FedTGP

- **ACL** can also be applied to other tasks and scenarios

$$\mathcal{L}_P^c = \sum_{i \in \mathcal{I}^t} -\log \frac{e^{-(\phi(P_i^c, \hat{P}^c) + \delta(t))}}{e^{-(\phi(P_i^c, \hat{P}^c) + \delta(t))} + \sum_{c'} e^{-\phi(P_i^c, \hat{P}^{c'})}}$$

③④ Data scarcity and privacy of vertical domain

- Vertical domain (e.g., **medical domain**) suffer from **data scarcity** and **privacy**
- Transfer **common knowledge** from large generative models to user models

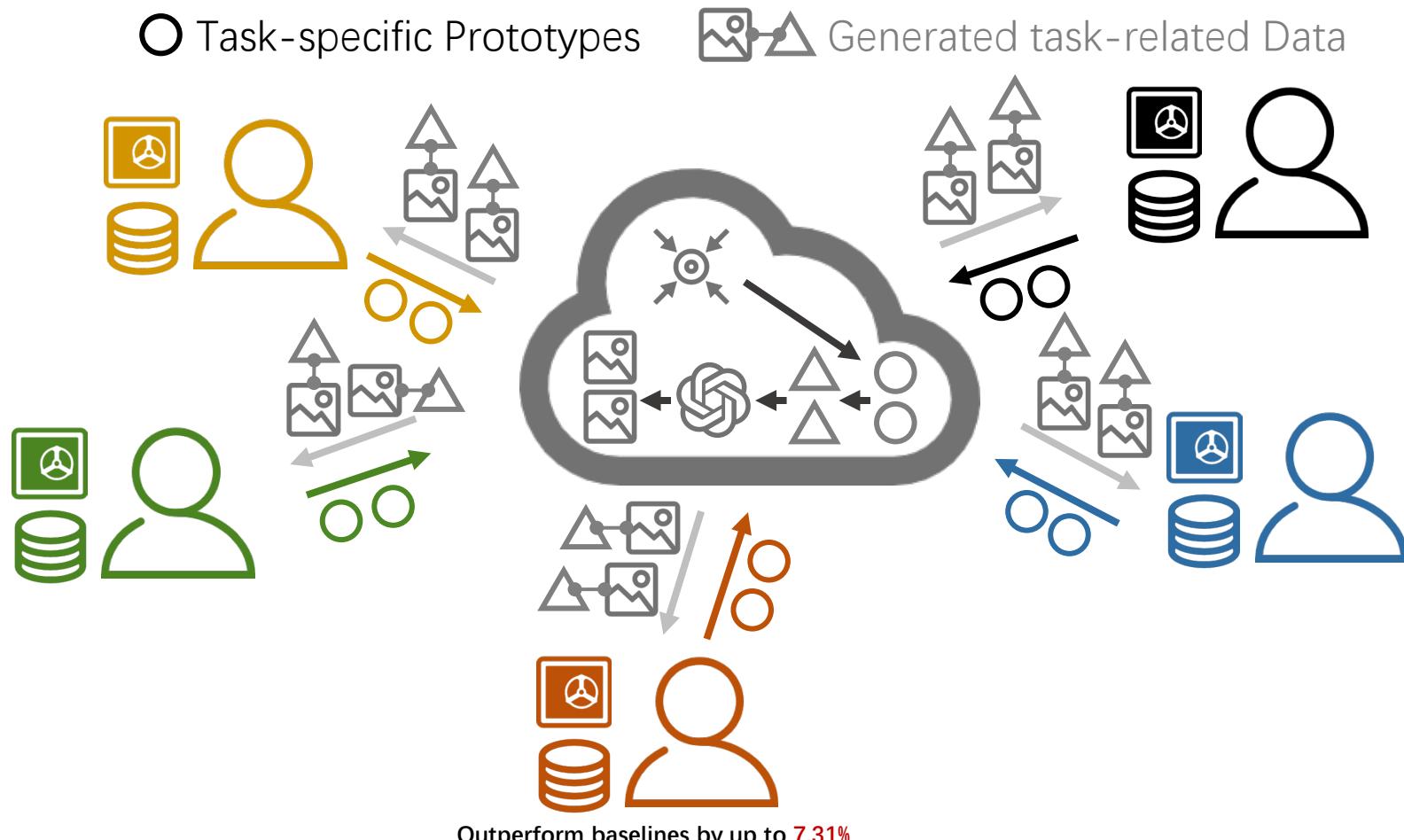


③④ Publications

- **[CVPR'24]** An Upload-Efficient Scheme for Transferring Knowledge From a Server-Side Pre-trained Generator to Clients in Heterogeneous Federated Learning.
- **How to obtain and transfer common and task-specific knowledge?**

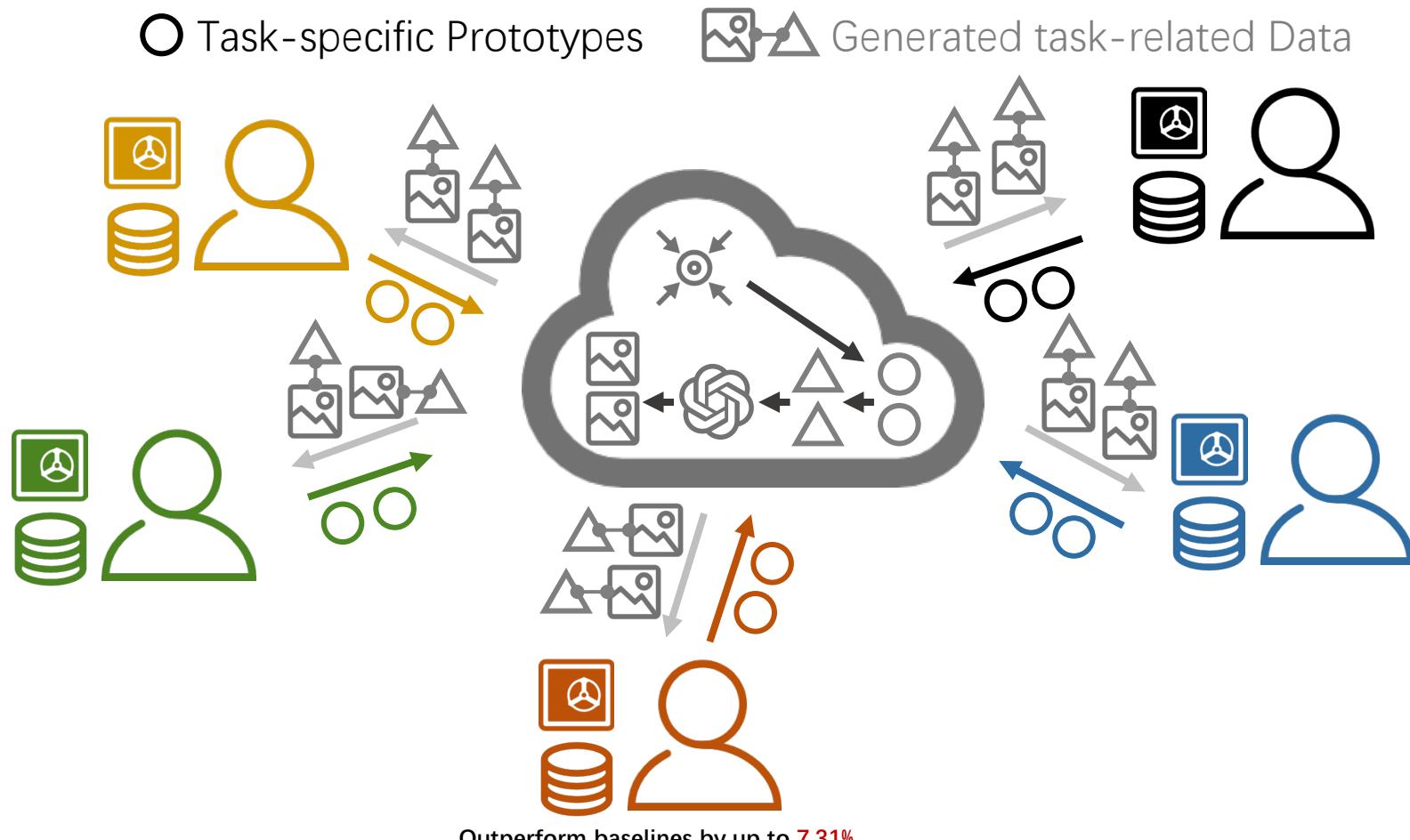
FedKTL

- Transfer **common knowledge** from the generator to clients
- Obtain **task-specific knowledge** from other clients



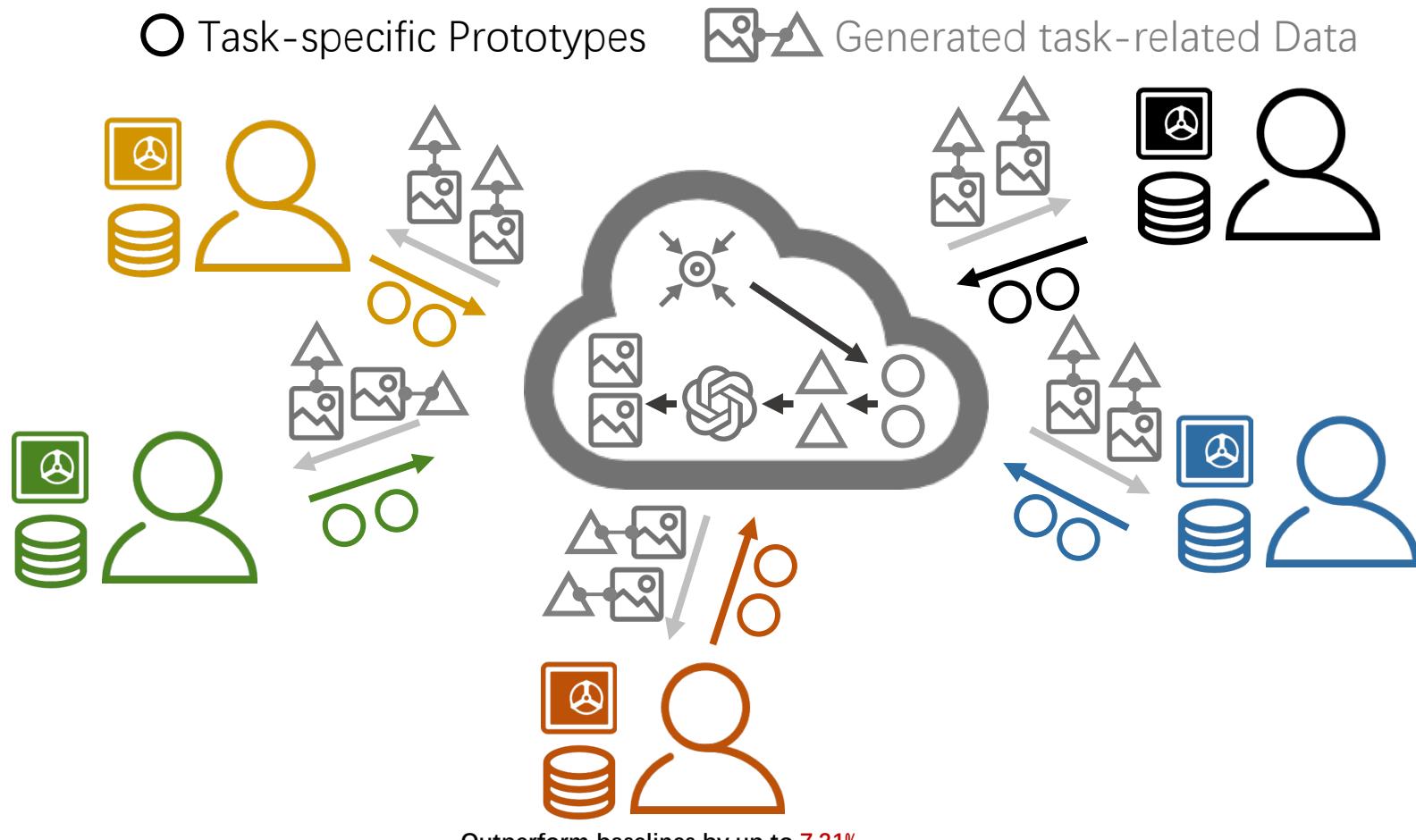
FedKTL

- Common knowledge: **generated images**
- Task-specific knowledge: **prototype vectors**



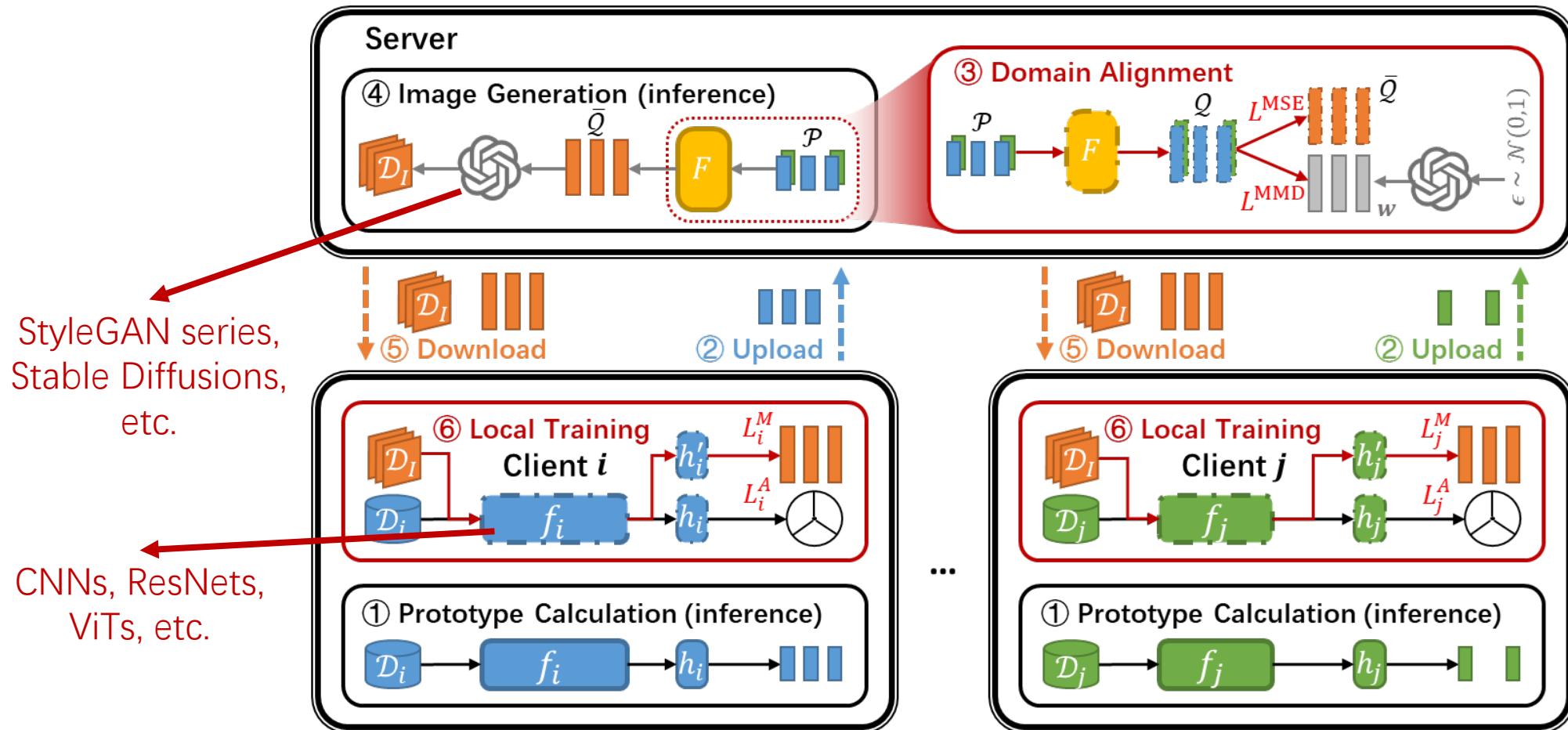
FedKTL

- Generated Images are **induced by** prototype vectors
- Image-vector pairs** are **task-related** data that **contain common knowledge**



FedKTL

- Transfer knowledge using an **additional supervised local task**
- Details:



FedKTL

- **One image per class** is sufficient for FedKTL
- FedKTL can **surpass** baselines (only using task-specific knowledge) **by a large margin**

Settings	Pathological Setting				Practical Setting			
	Datasets	Cifar10	Cifar100	Flowers102	Tiny-ImageNet	Cifar10	Cifar100	Flowers102
LG-FedAvg	86.82±0.26	57.01±0.66	58.88±0.28	32.04±0.17	84.55±0.51	40.65±0.07	45.93±0.48	24.06±0.10
FedGen	82.83±0.65	58.26±0.36	59.90±0.15	29.80±1.11	82.55±0.49	38.73±0.14	45.30±0.17	19.60±0.08
FedGH	86.59±0.23	57.19±0.20	59.27±0.33	32.55±0.37	84.43±0.31	40.99±0.51	46.13±0.17	24.01±0.11
FML	87.06±0.24	55.15±0.14	57.79±0.31	31.38±0.15	85.88±0.08	39.86±0.25	46.08±0.53	24.25±0.14
FedKD	87.32±0.31	56.56±0.27	54.82±0.35	32.64±0.36	86.45±0.10	40.56±0.31	48.52±0.28	25.51±0.35
FedDistill	87.24±0.06	56.99±0.27	58.51±0.34	31.49±0.38	86.01±0.31	41.54±0.08	49.13±0.85	24.87±0.31
FedProto	83.39±0.15	53.59±0.29	55.13±0.17	29.28±0.36	82.07±1.64	36.34±0.28	41.21±0.22	19.01±0.10
FedKTL	88.43±0.13	62.01±0.28	64.72±0.62	34.74±0.17	87.63±0.07	46.94±0.23	53.16±0.08	28.17±0.18

Table 1. The test accuracy (%) on four datasets in the pathological and practical settings using HtFE₈.

FedKTL

- FedKTL can **adapt to various generators** that were pre-trained using various datasets
- The **semantics of the generated images** can be different from clients' data



(a) Client #1



(b) AFHQv2



(c) Benches



(d) FFHQ-U



(e) WikiArt

Generators pre-trained on different image datasets

FedKTL

- FedKTL can **adapt to various generators** that were pre-trained using various datasets
- The **semantics of the generated images** can be different from clients' data

	$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.5$
AFHQv2	26.82±0.32	27.05±0.26	26.32±0.52
Bench	27.71±0.25	28.36±0.42	27.56±0.50
FFHQ-U	27.28±0.23	27.21±0.35	26.59±0.47
WikiArt	27.37±0.51	27.48±0.33	27.30±0.15

Table 6. The test accuracy (%) on Tiny-ImageNet in the practical setting using HtFE₈ with different pre-trained StyleGAN3s, which are represented by the names of the pre-training datasets.

FedKTL

- **Knowledge transfer scheme (KTL)** is also applicable in scenarios with **only one edge client**.
- The **cloud-edge** scenario

Settings	100-way 23-shot	100-way 9-shot	100-way 2-shot
Client Data	12.53 ± 0.39	7.55 ± 0.41	4.44 ± 1.66
Our KTL	13.02 ± 0.43	8.88 ± 0.62	8.76 ± 2.25
Improvement	0.49	1.33	4.32
Improvement Ratio	3.91%	17.61%	97.29%

Table 9. The test accuracy (%) with Cifar100's subsets on a single client using a small model *i.e.*, the 4-layer CNN.

Feel free to contact me!

Home page: <https://github.com/TsingZ0>



Thanks!