Federated Learning and Its Extension With Large Pre-trained Generators

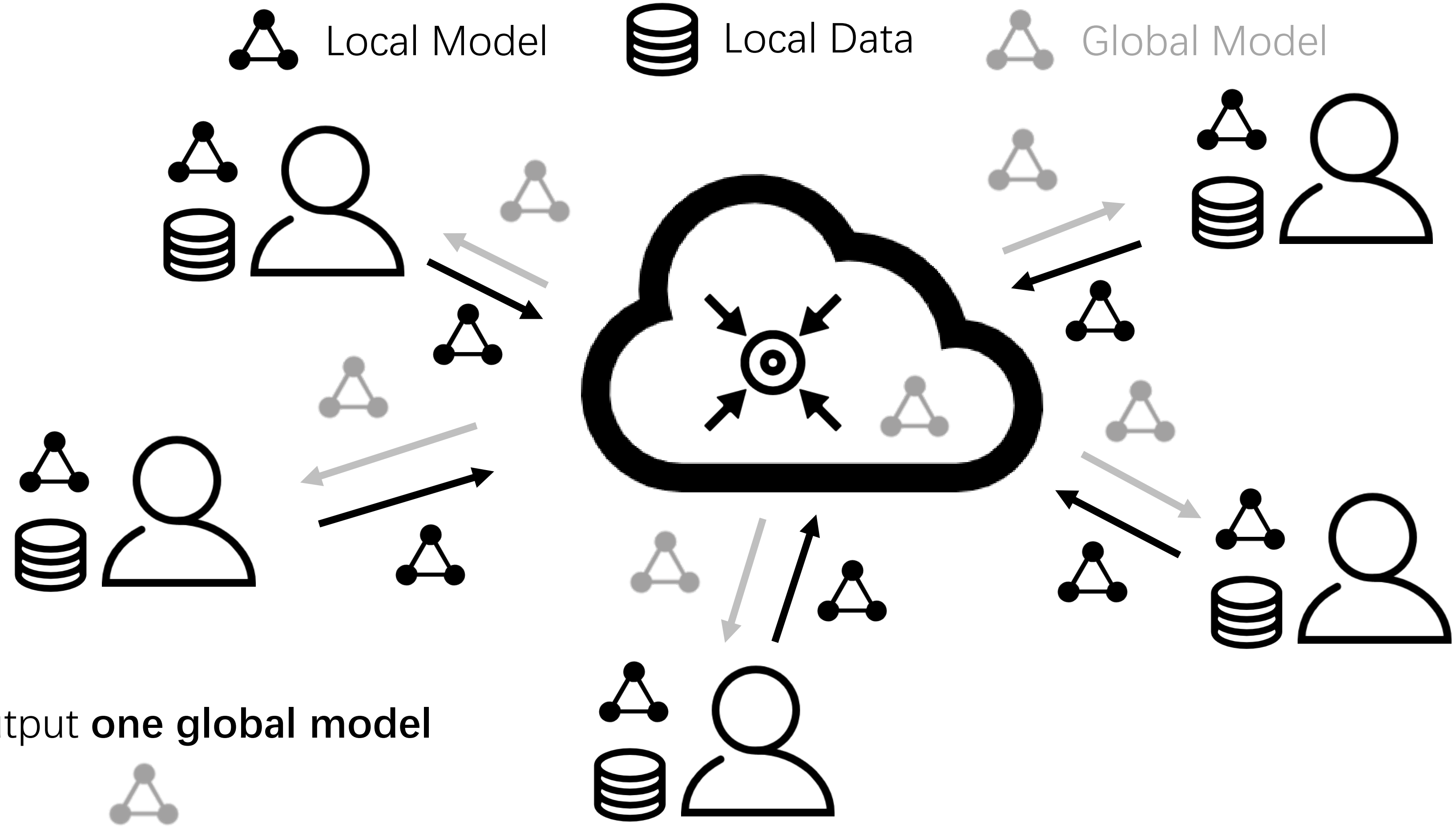
- **Name:** Jianqing Zhang
- **Age:** 26
- **Ph.D.:** Shanghai Jiao Tong University
- **M.S.:** Shanghai Jiao Tong University
- **Collaborations:**
  - Tsinghua University
  - Queen's University Belfast
  - Louisiana State University

Content

- **Research interests**
  - Federated learning, transfer learning, recommender systems
- **Projects**
  - PFLlib (1000+ stars, 200+ forks), HtFL, FL-IoT, etc.
- **Featured publications (first author)**
  - Stage ① [personalized federated learning]:
    - AAAI’23, KDD’23, ICCV’23, NeurIPS’23, PFLlib’paper
  - Stage ② [heterogeneous federated learning]:
    - AAAI’24, CVPR’24

Federated Learning (FL)

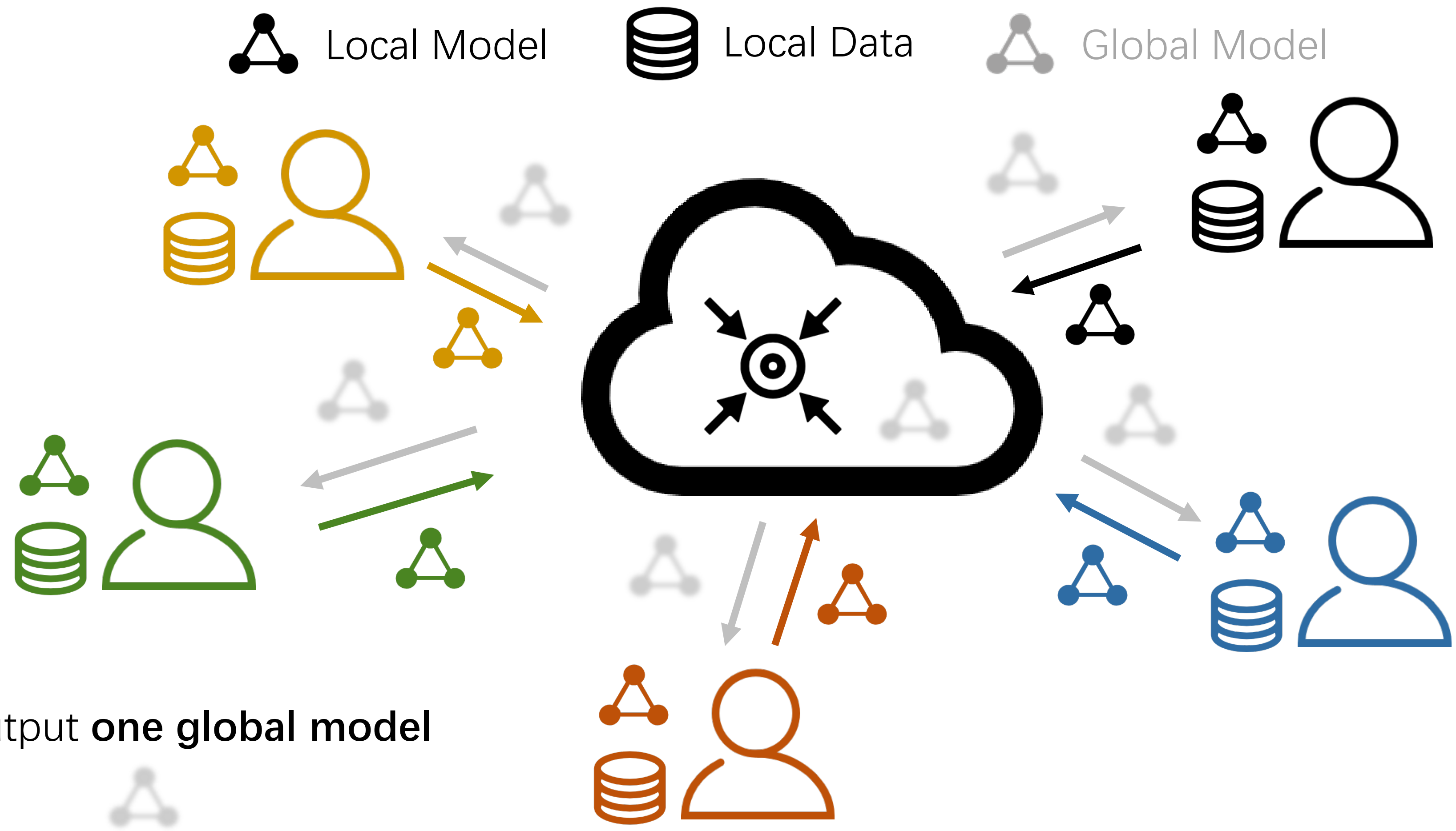
- A **collaborative-learning** and **privacy-preserving** technique
- Learn an AI model among clients by **only sharing models** with the server.



Finally output **one global model**

① Data Heterogeneity Issue in FL

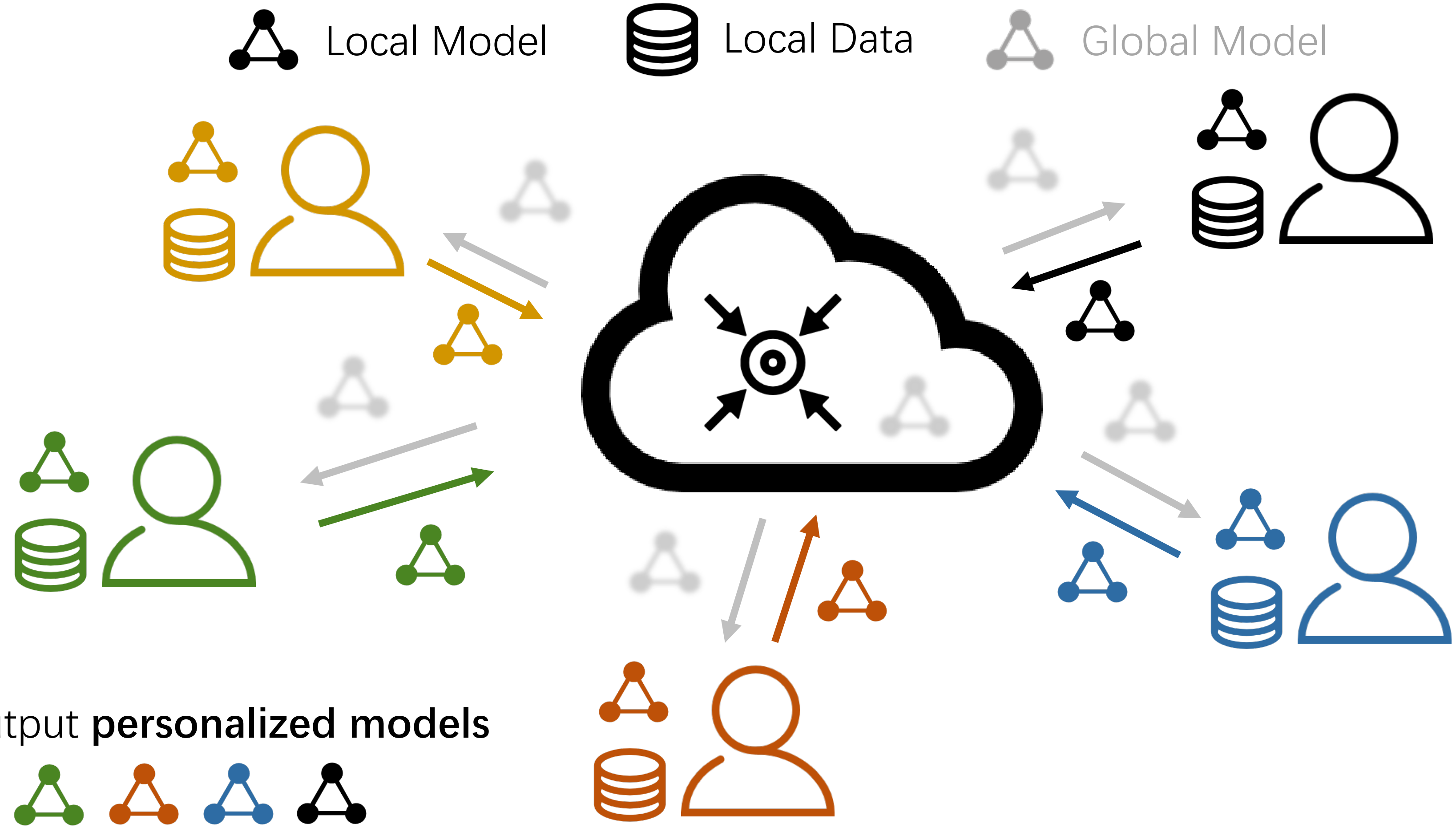
- Data is **generated by different clients** and forbidden to be manipulated
- Each client has **personalized preferences**



Finally output **one global model**

① Data Heterogeneity Issue in FL

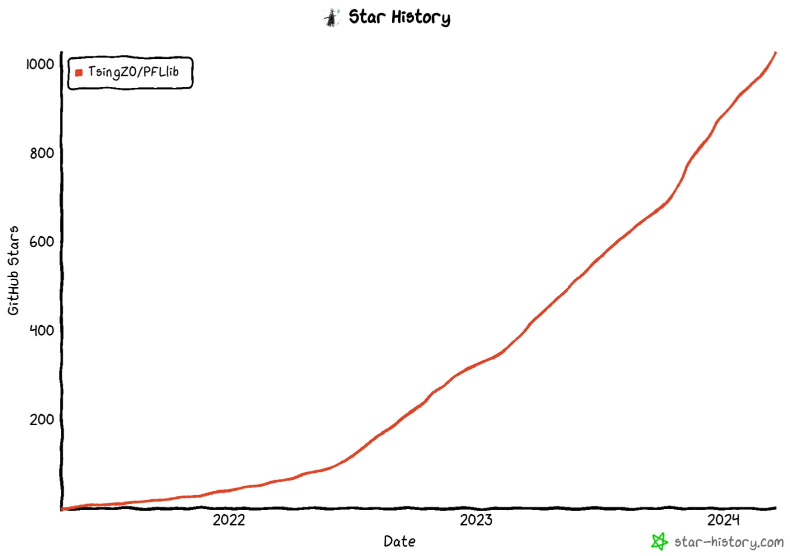
- Data is **generated by different clients** and forbidden to be manipulated
- Each client has **personalized preferences**



Finally output **personalized models**

① PFLlib: Personalized FL (pFL) Algorithm Library

- **Beginner-friendly**
- **Comprehensive (34 pFLs)**
- **Popular (1000+ stars)**
- **…**



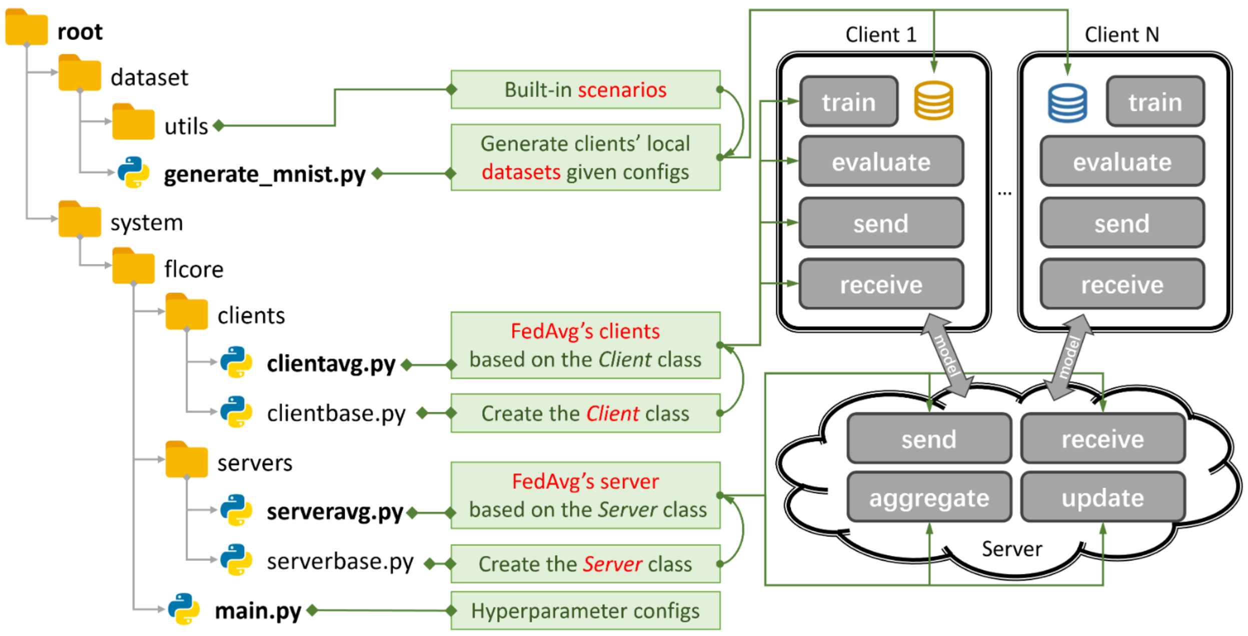


Figure 1: An Example for FedAvg. You can create a scenario using `generate_DATA.py` and run an algorithm using `main.py`, `clientNAME.py`, and `serverNAME.py`.

We expose this user-friendly algorithm library (with an integrated evaluation platform) for beginners who intend to start federated learning (FL) study.

- ***34 traditional FL (tFL) or personalized FL (pFL) algorithms, 3 scenarios, and 20 datasets.***
- Some **experimental results** are avalible 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.*

https://github.com/TsingZ0/PFLlib

① Featured 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.

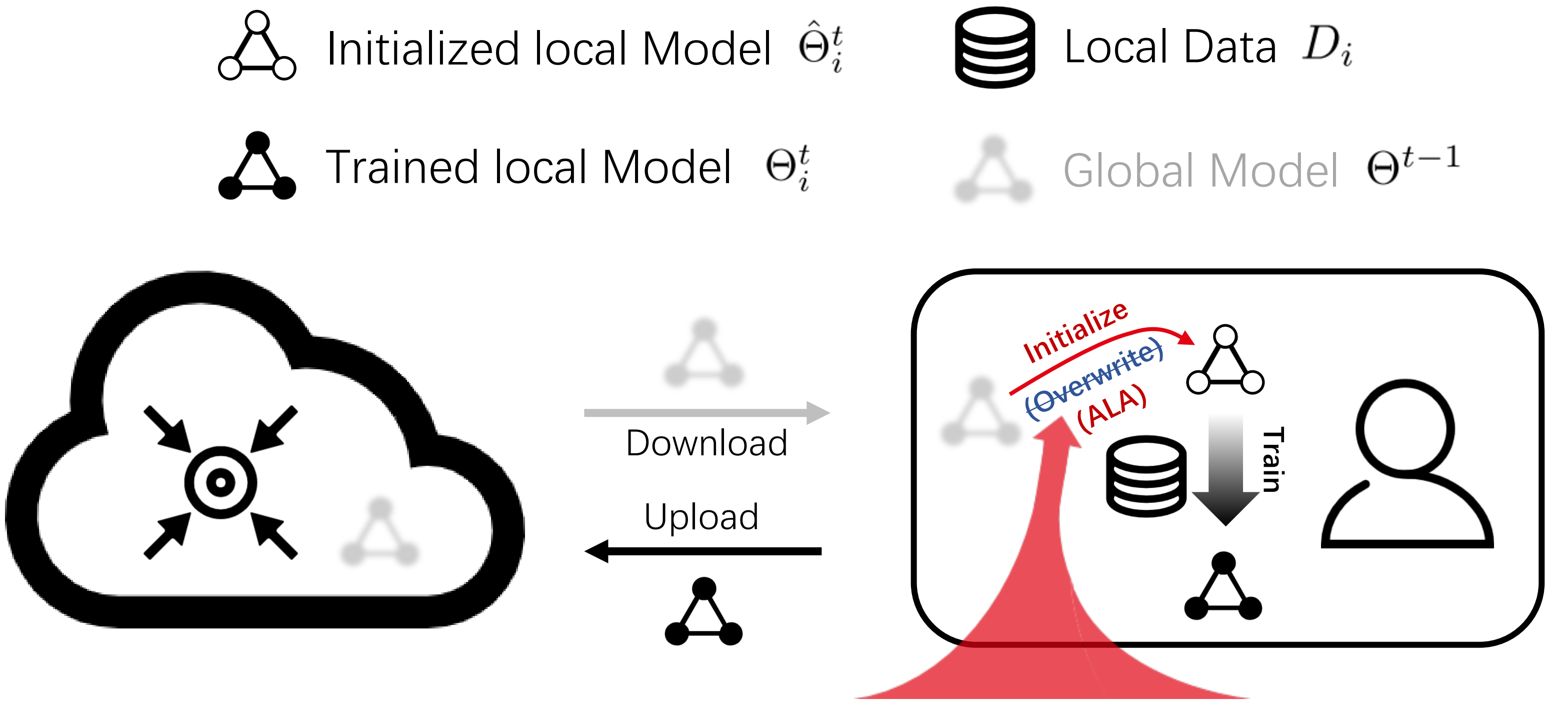
- **How can we consider both generalization and personalization?**

① Featured 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

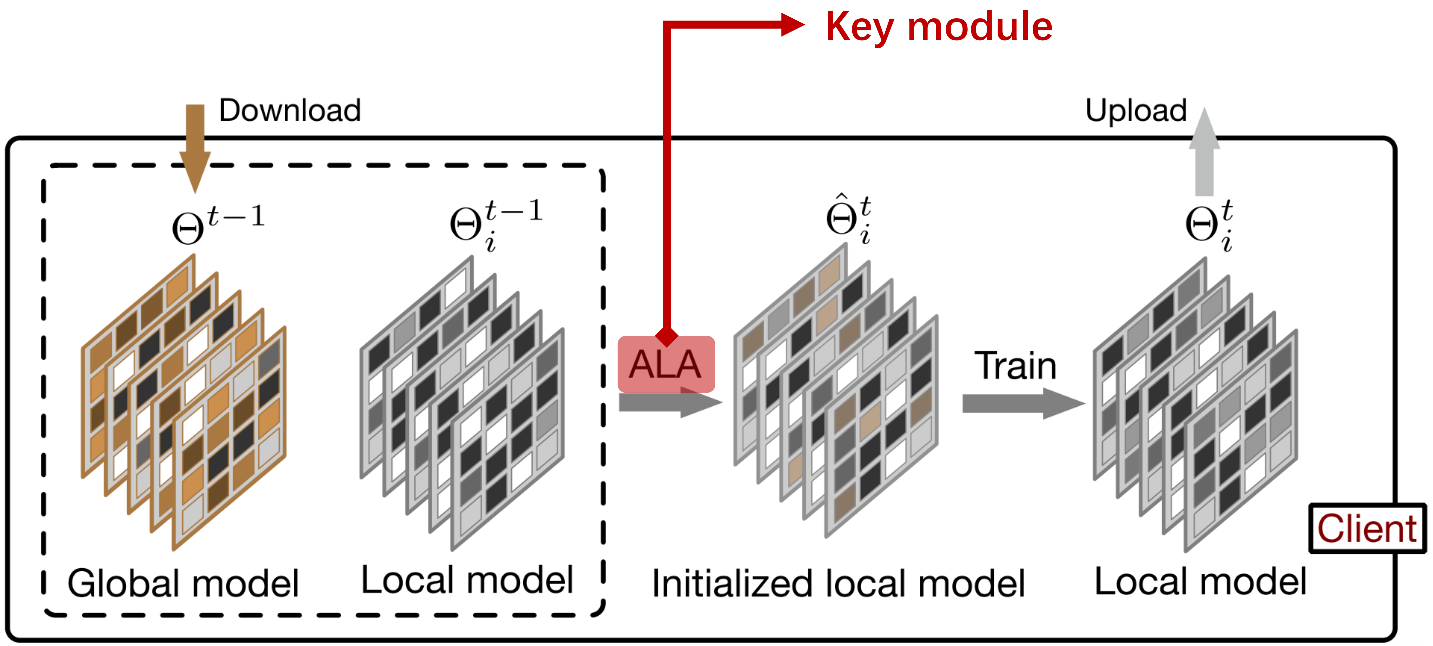
- Original workflow in FL
  - Both the **desired** and undesired information exist in the global model, resulting in poor generalization ability



Do not modify other FL process

FedALA

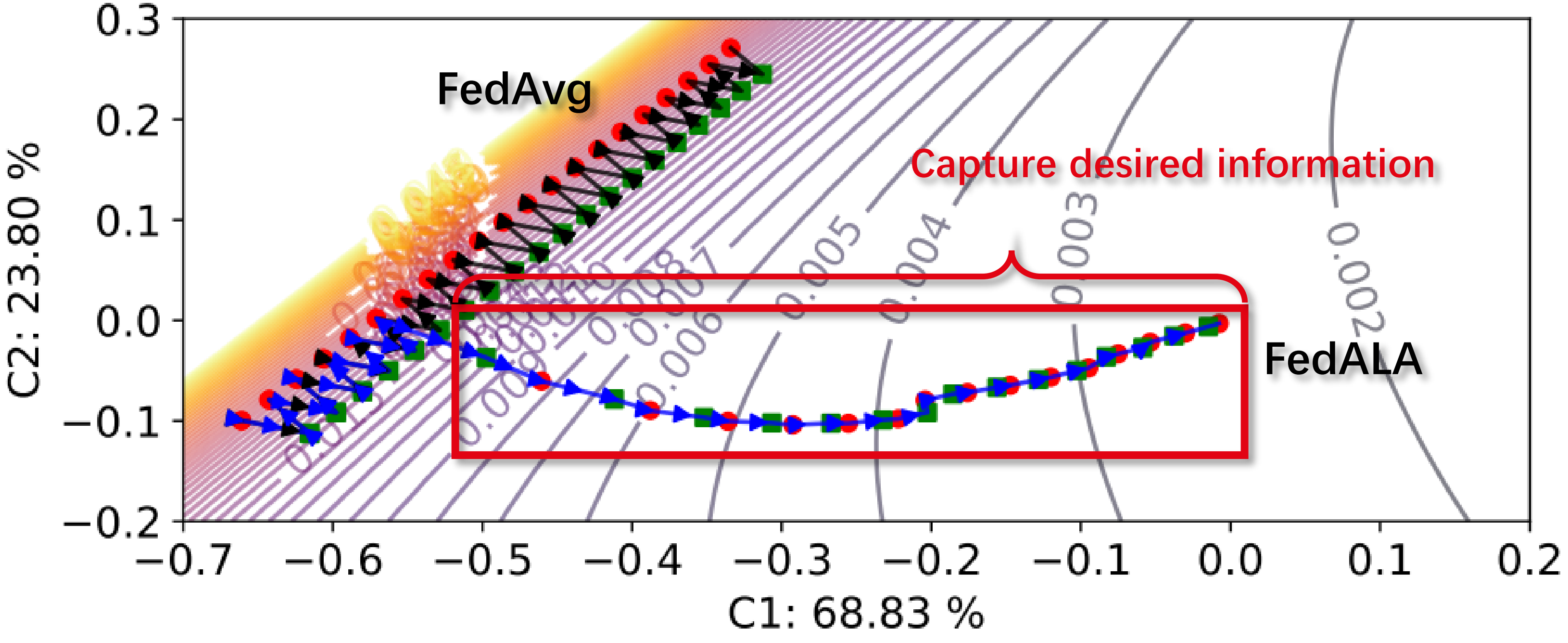
- **Extract each client's desired information from the global model** that facilitates local training
- **Adaptively aggregate** the information in the global and local model for initialization



Workflow on the client in one iteration

FedALA

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



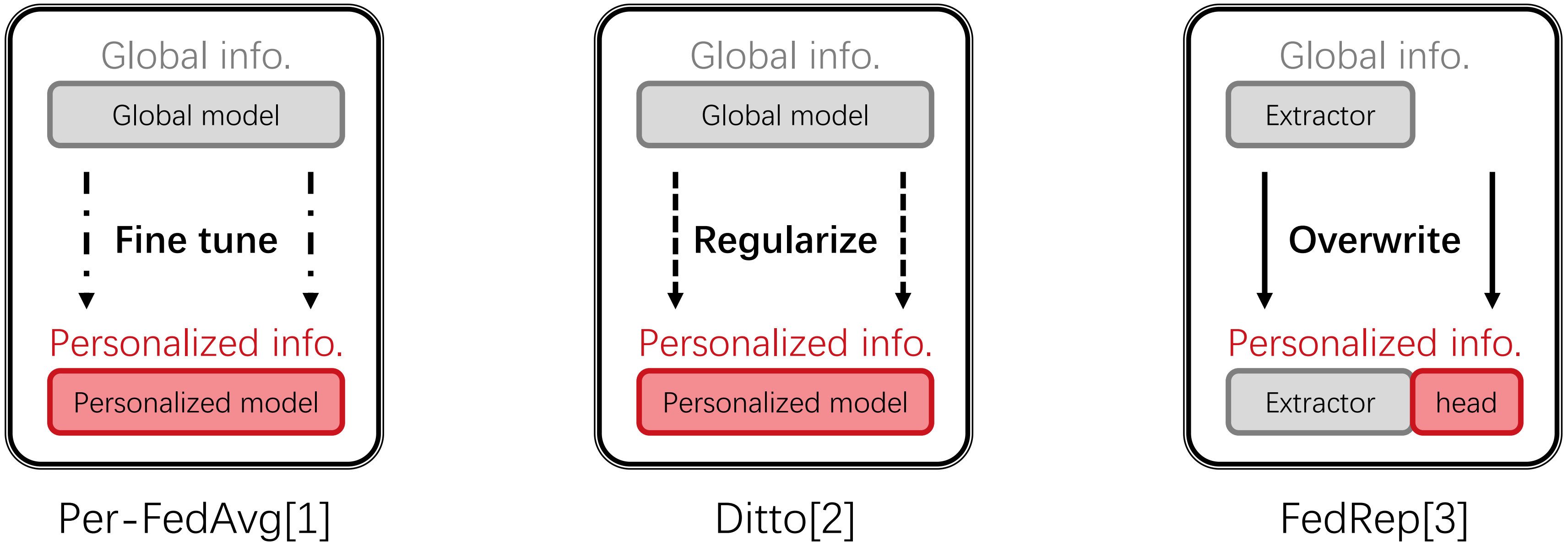
2D visualization of local learning trajectory

① Featured 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.

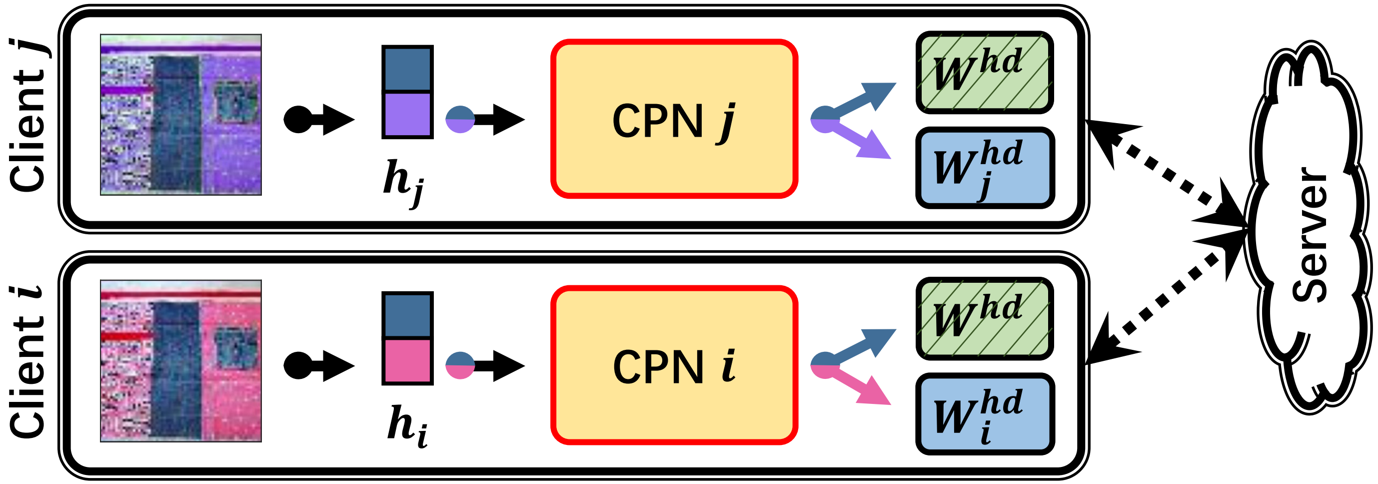


- They only focus on model parameters, but **ignore the source of information: data.**

[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

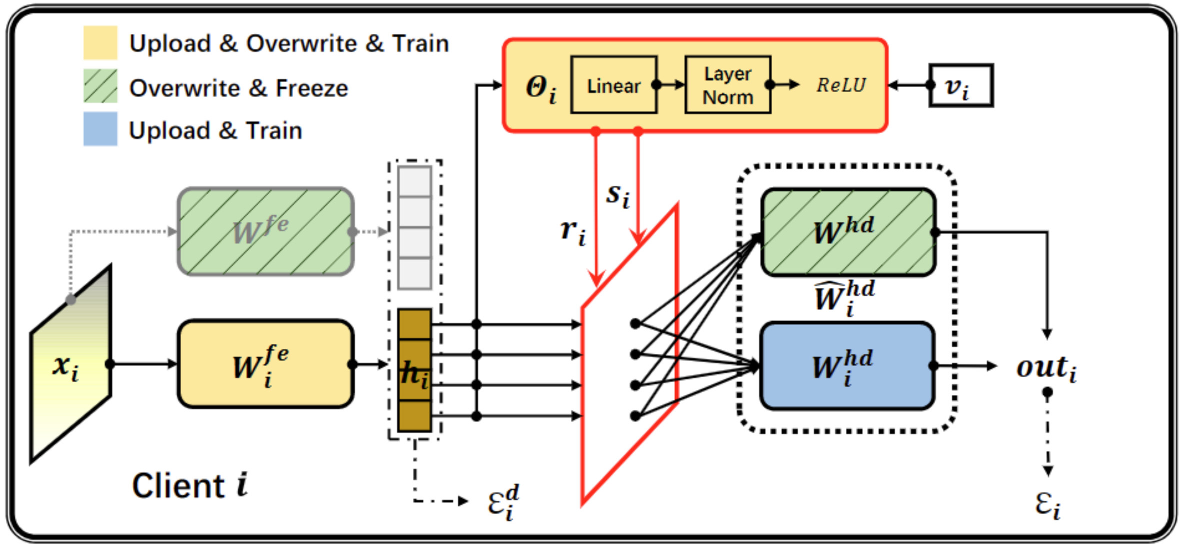
- We separate feature information via an *auxiliary* **Conditional Policy Network (CPN).**
  - Generate sample-specific policy
  - End-to-end training together with the client model
  - Lightweight (e.g., 4.67% parameters of ResNet-18)



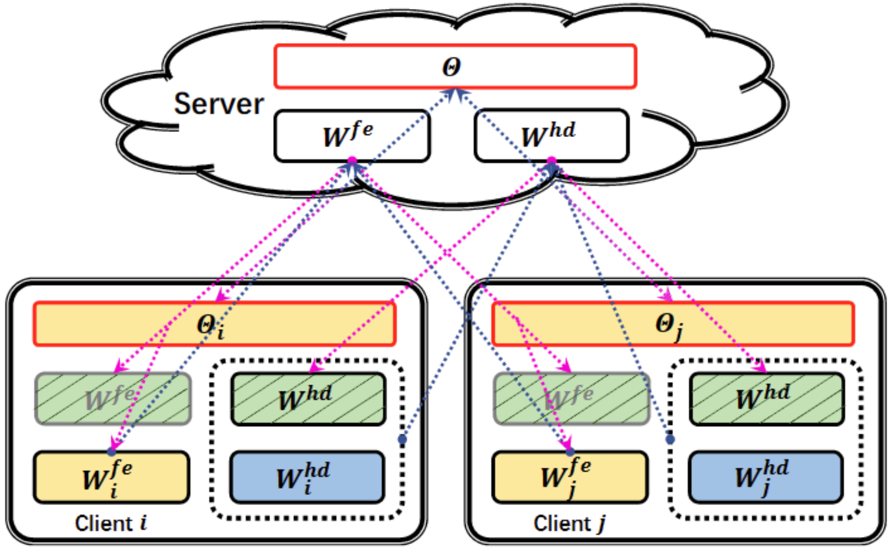
- We utilize global and personalized information via global and personalized heads.

FedCP

- Architecture



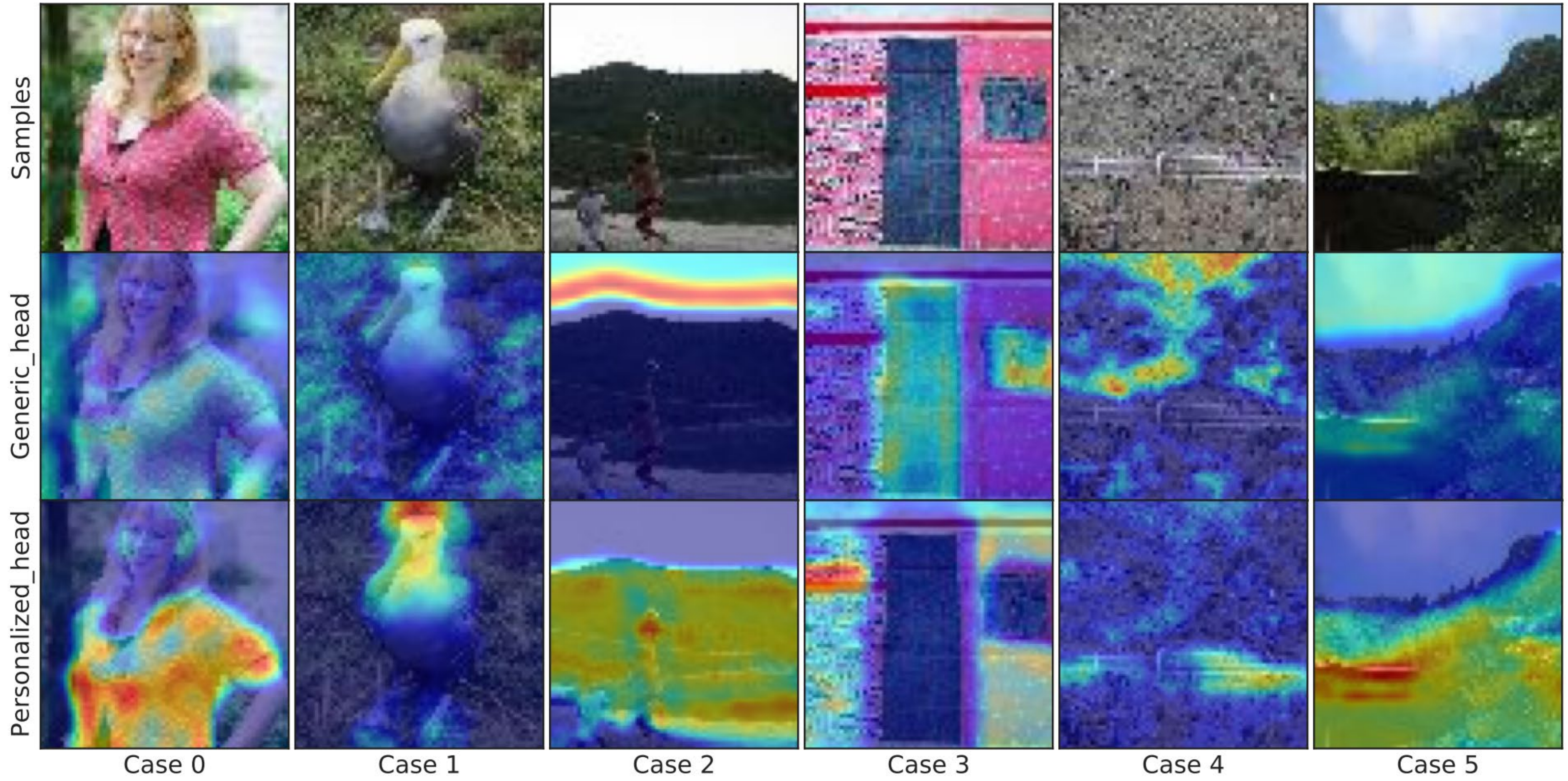
Data flow in the personalized model



Upload and download stream

FedCP

- **Separating Feature Information**



Six samples from the Tiny-ImageNet dataset

① Featured 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

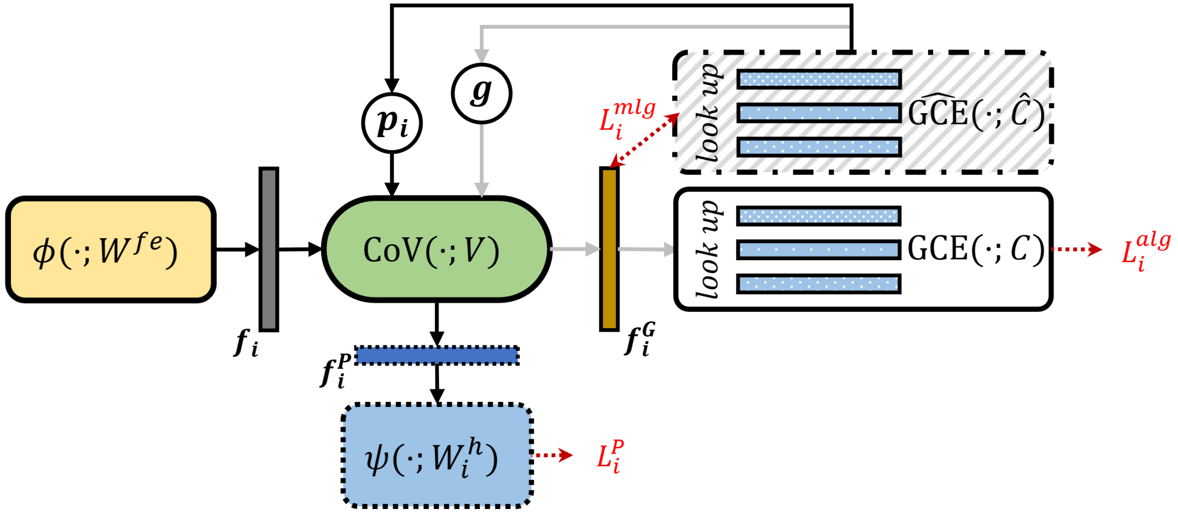
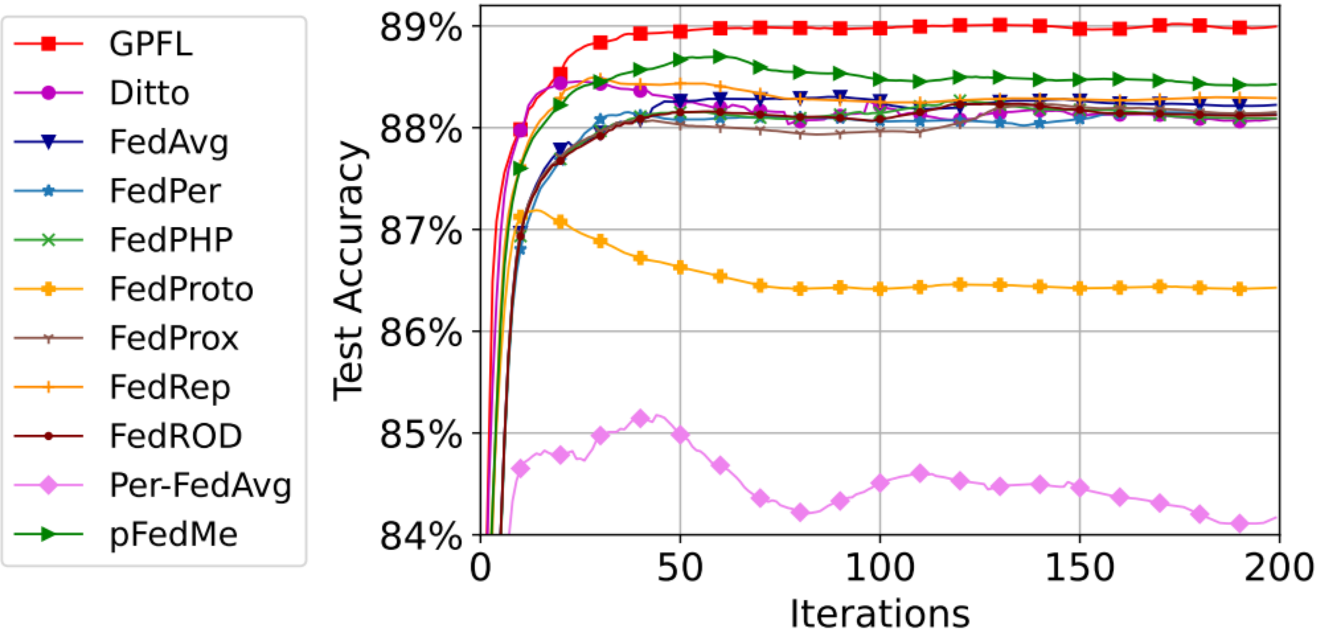


Illustration of client modules and data flow between them

GPFL

- Address the **overfitting** issue in pFL



Test accuracy curves in the feature shift setting

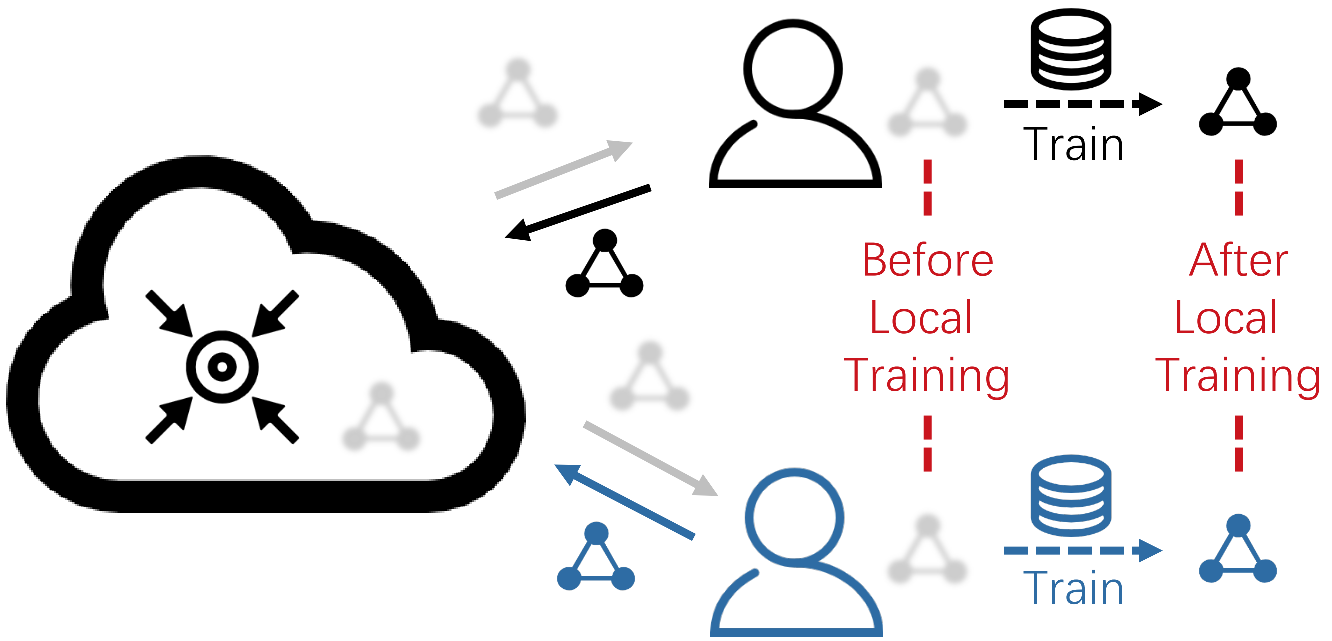
Outperform baselines by up to 8.99%

① Featured 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.

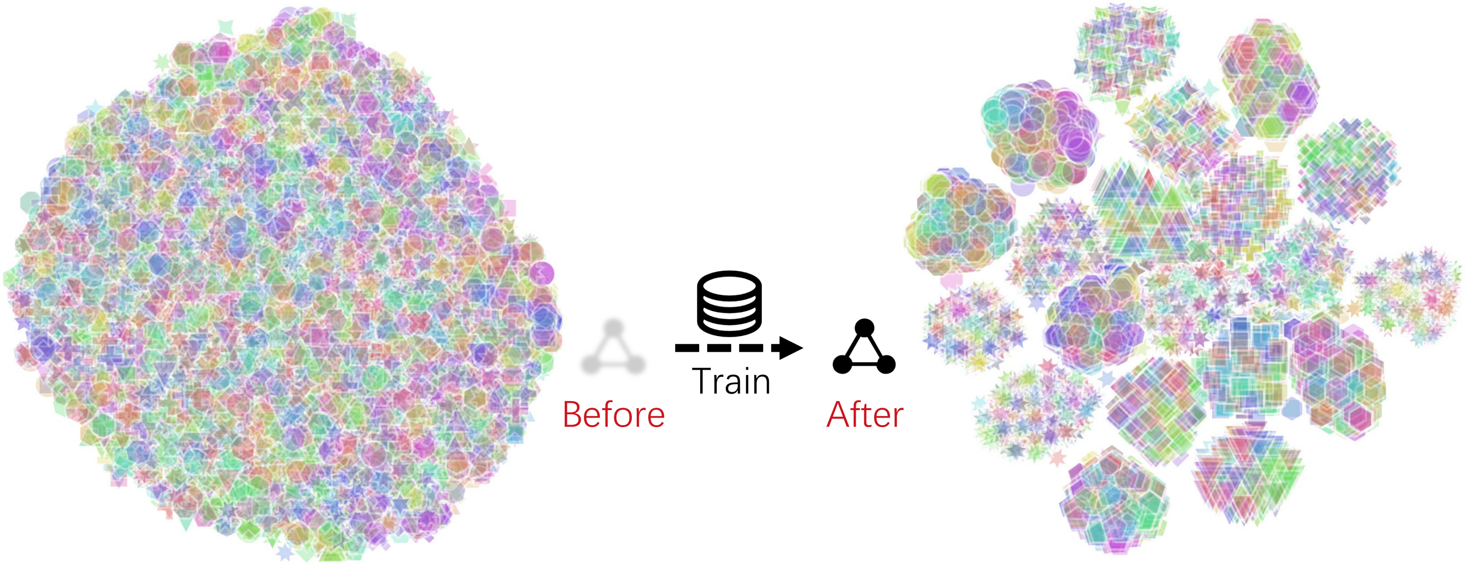
Local training

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



Representation bias phenomenon

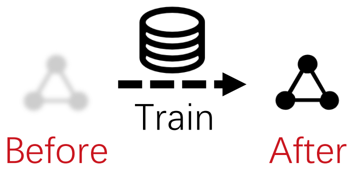
- After local training, the feature representations are **biased** to client-specific domains



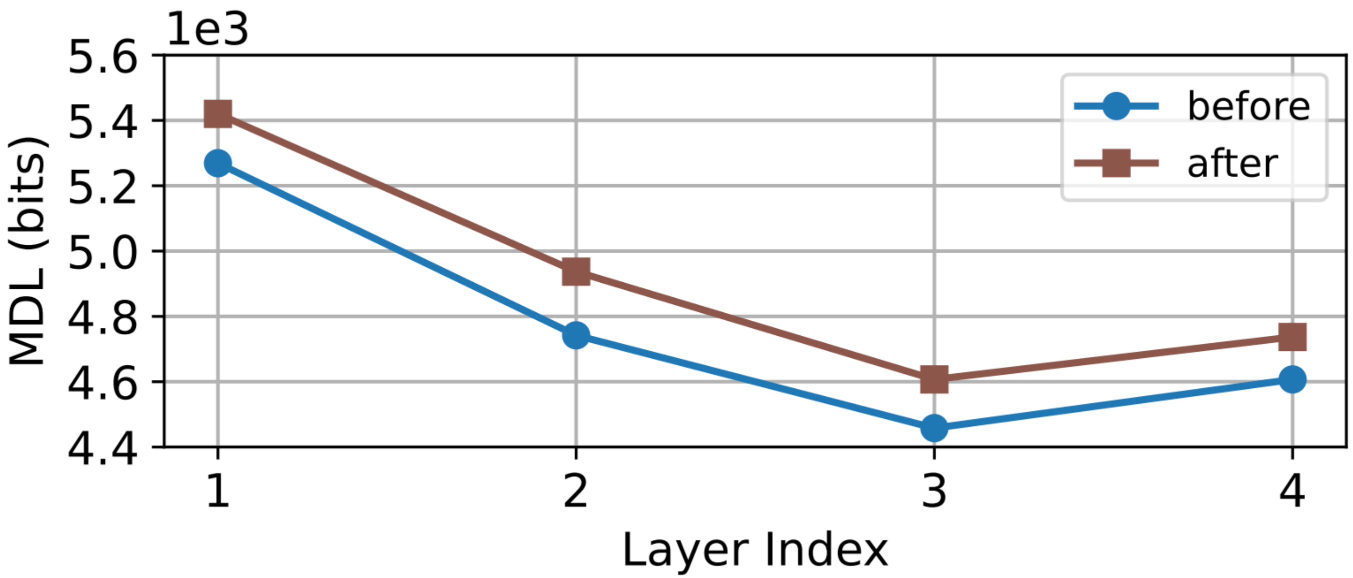
(a) Before local training (b) After local training

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

Representation degeneration phenomenon



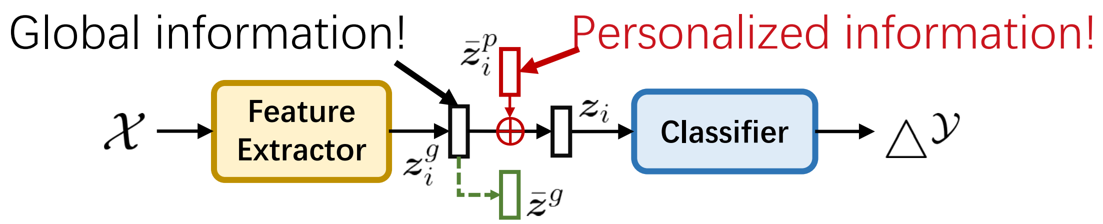
- At the same time, representations' quality is also **degenerated**



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 $1 - \delta$:*

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

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 $\boxed{d_{\mathcal{H}}(\tilde{\mathcal{U}}_i^g, \tilde{\mathcal{U}}^g) \leq d_{\mathcal{H}}(\tilde{\mathcal{U}}_i, \tilde{\mathcal{U}})}$. $\tilde{\mathcal{U}}_i^g$ 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 $\boxed{1}$. Given a translation transformation function $\mathtt{PRBM}: \mathcal{Z} \mapsto \mathcal{Z}$ 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^*, $\mathcal{F}' = \mathtt{PRBM} \circ \mathcal{F}^g : \mathcal{X} \mapsto \mathcal{Z}$, then for every $h' \in \mathcal{H}$, with probability at least $1 - \delta$:*

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

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

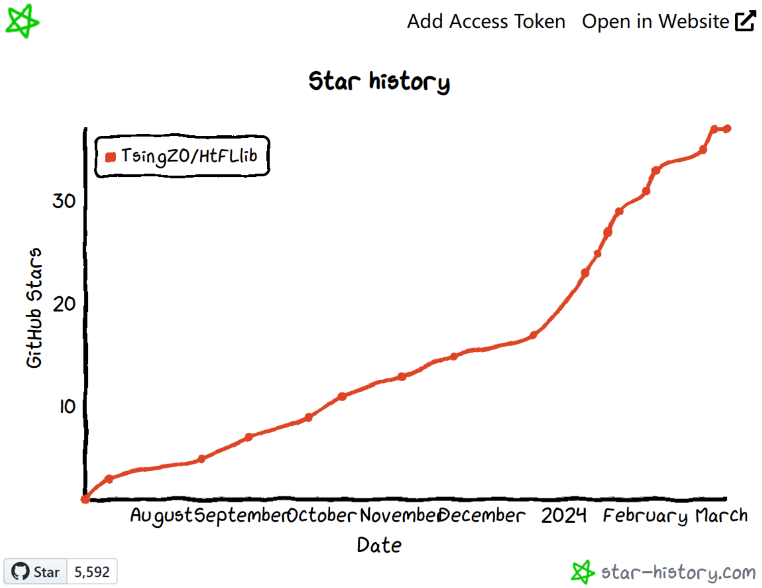
Please refer to our paper for proofs.

② Model heterogeneity issue in FL

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

② HtFLlib: HtFL Algorithm Library

- **Burgeoning**
- **Beginner-friendly**
- **Data-free**
- **Comprehensive**
- **…**



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.

- **Local** — Each client trains its model locally without federation.
- **FedDistill** — Federated Knowledge Distillation *2020*
- **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*

https://github.com/TsingZ0/HtFLlib

② Featured 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.

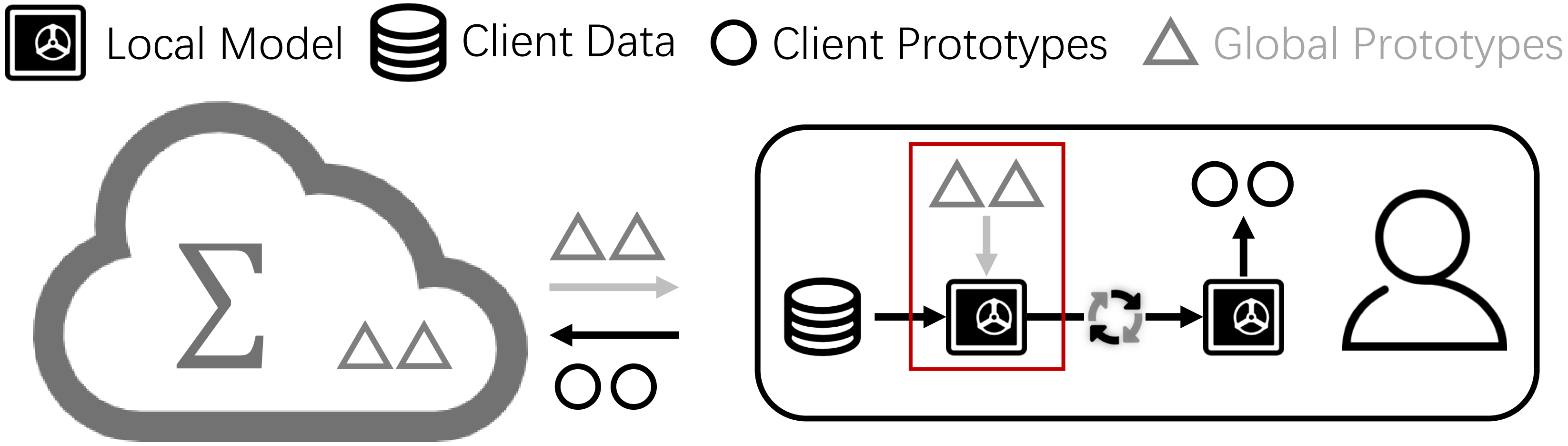
- **How can knowledge be shared and aggregated to benefit participants?**

② Featured 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.

FedProto: share prototypes (class representatives)

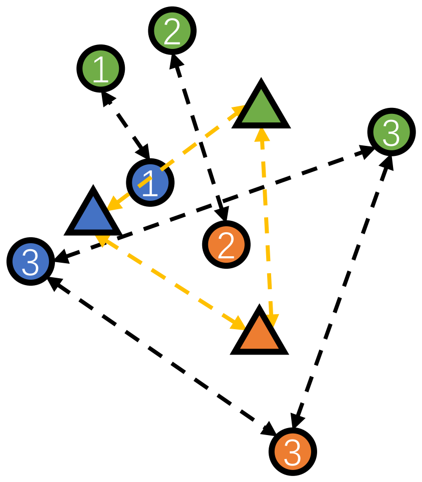
- Share **client prototypes** with the server
- Aggregate client prototypes to generate **global prototypes**
- Train client models with both client data and global prototypes



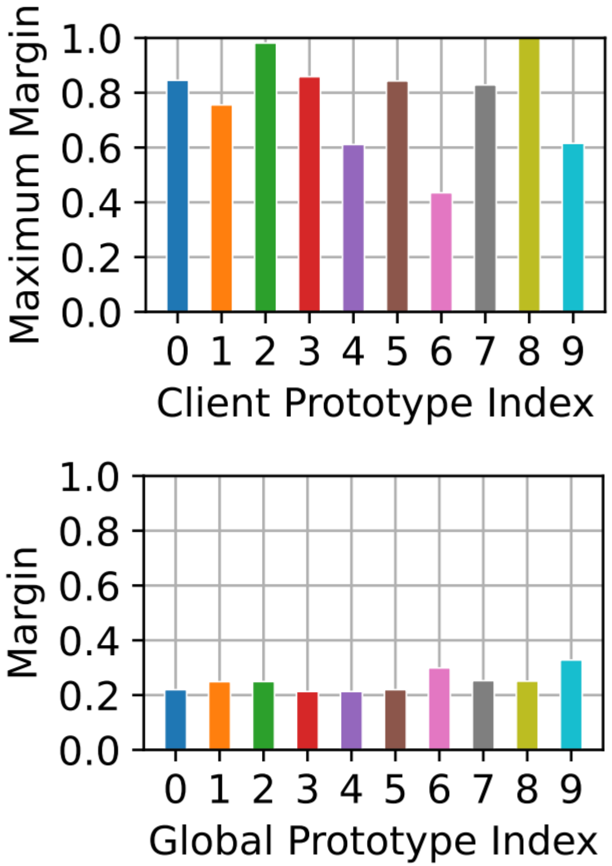
[1] Tan Y, Long G, Liu L, et al. Fedproto: Federated prototype learning across heterogeneous clients. AAAI 2022.

Issues of FedProto

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



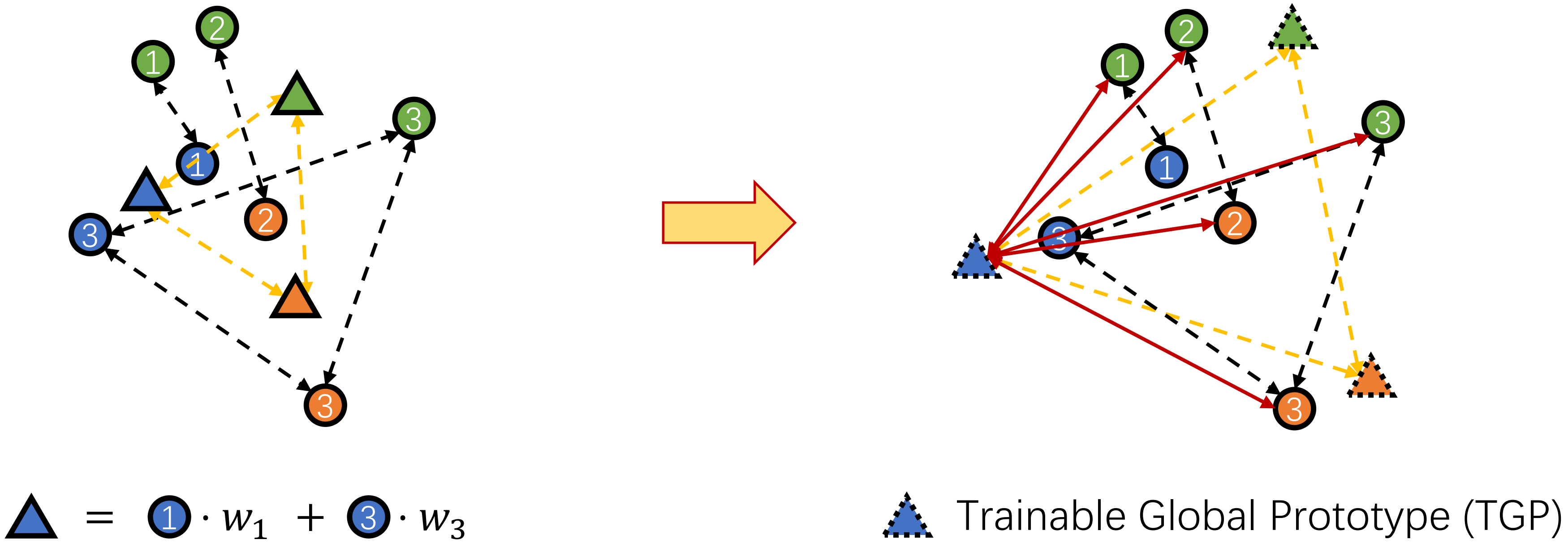
$$\triangle = ① \cdot w_1 + ③ \cdot w_3$$



[1] Tan Y, Long G, Liu L, et al. Fedproto: Federated prototype learning across heterogeneous clients. AAAI 2022.

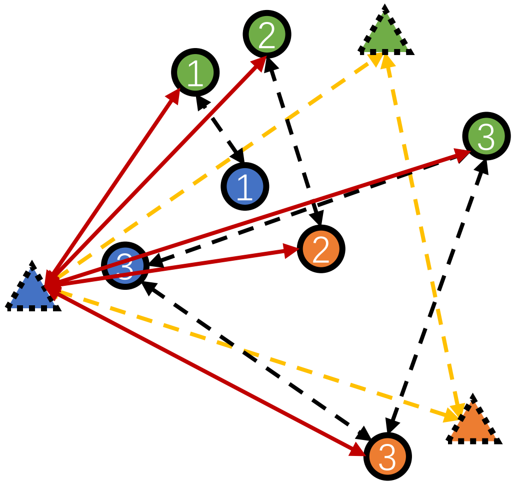
FedTGP

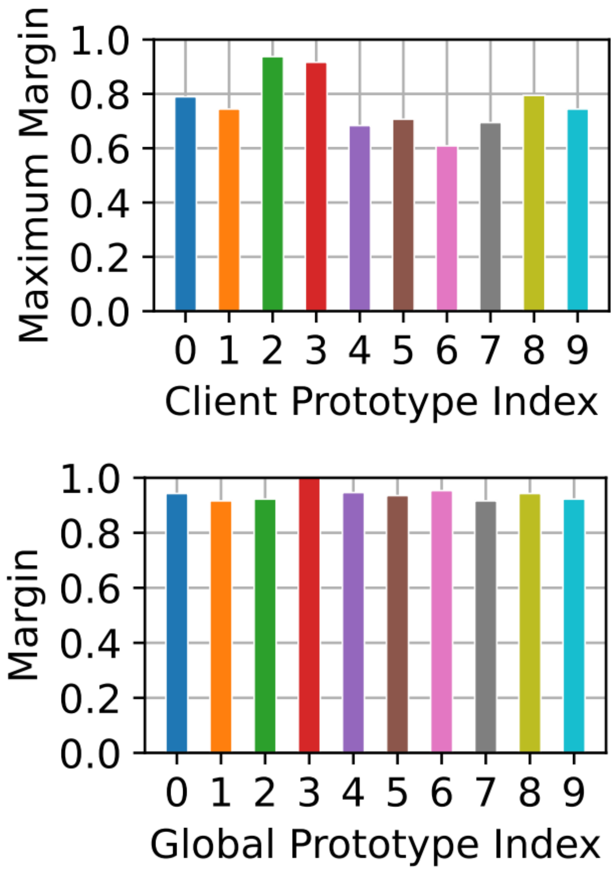
- Remove weighted-averaging
- Consider the uploaded client prototypes as data
- **Enlarge** the global prototype margin



FedTGP

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





Outperform baselines by up to 9.08%

FedTGP

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

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

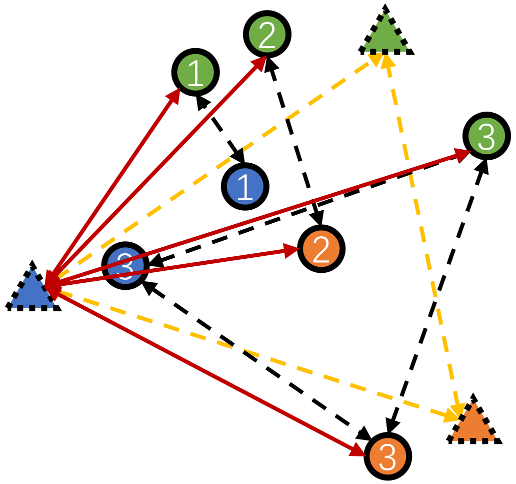
$$\mathcal{L}_P^c = \sum_{i \in \mathcal{I}^t} -\log \frac{e^{-(\phi(P_i^c, \hat{P}^c)+\delta(t))}}{e^{-(\phi(P_i^c, \hat{P}^c)+\delta(t))} + \sum_{c'} e^{-\phi(P_i^c, \hat{P}^{c'})}}$$

$$\delta(t) = \min(\max_{c\in[C], c'\in[C], c\neq c'} \phi(Q_t^c, Q_t^{c'}), \tau),$$

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

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

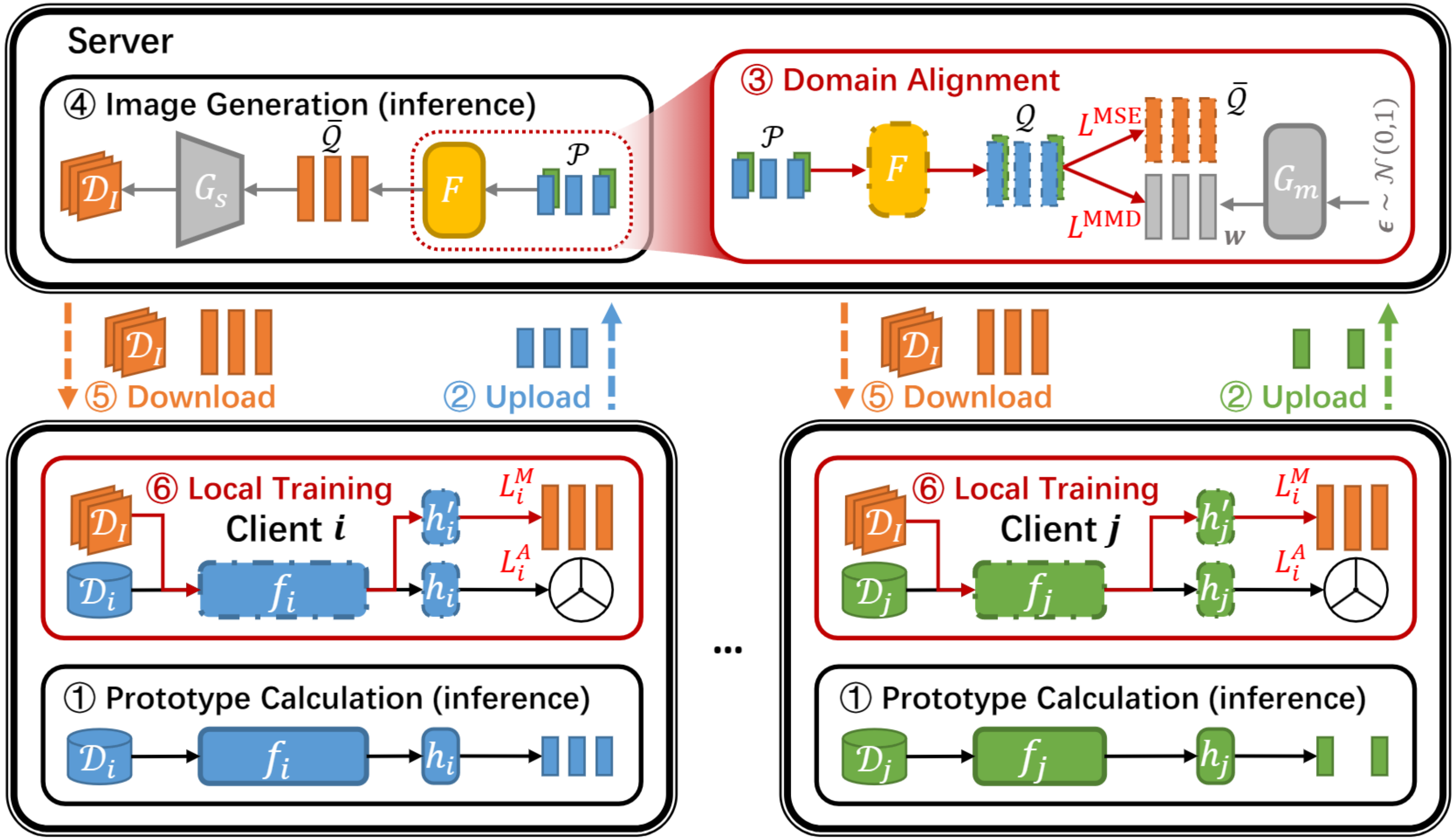
$$\mathcal{L}_P^c = \sum_{i \in \mathcal{I}^t} -\log \frac{e^{-(\phi(P_i^c, \hat{P}^c) + \delta(t))}}{e^{-(\phi(P_i^c, \hat{P}^c) + \delta(t))} + \sum_{c'} e^{-\phi(P_i^c, \hat{P}^{c'})}}$$

② Featured 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.

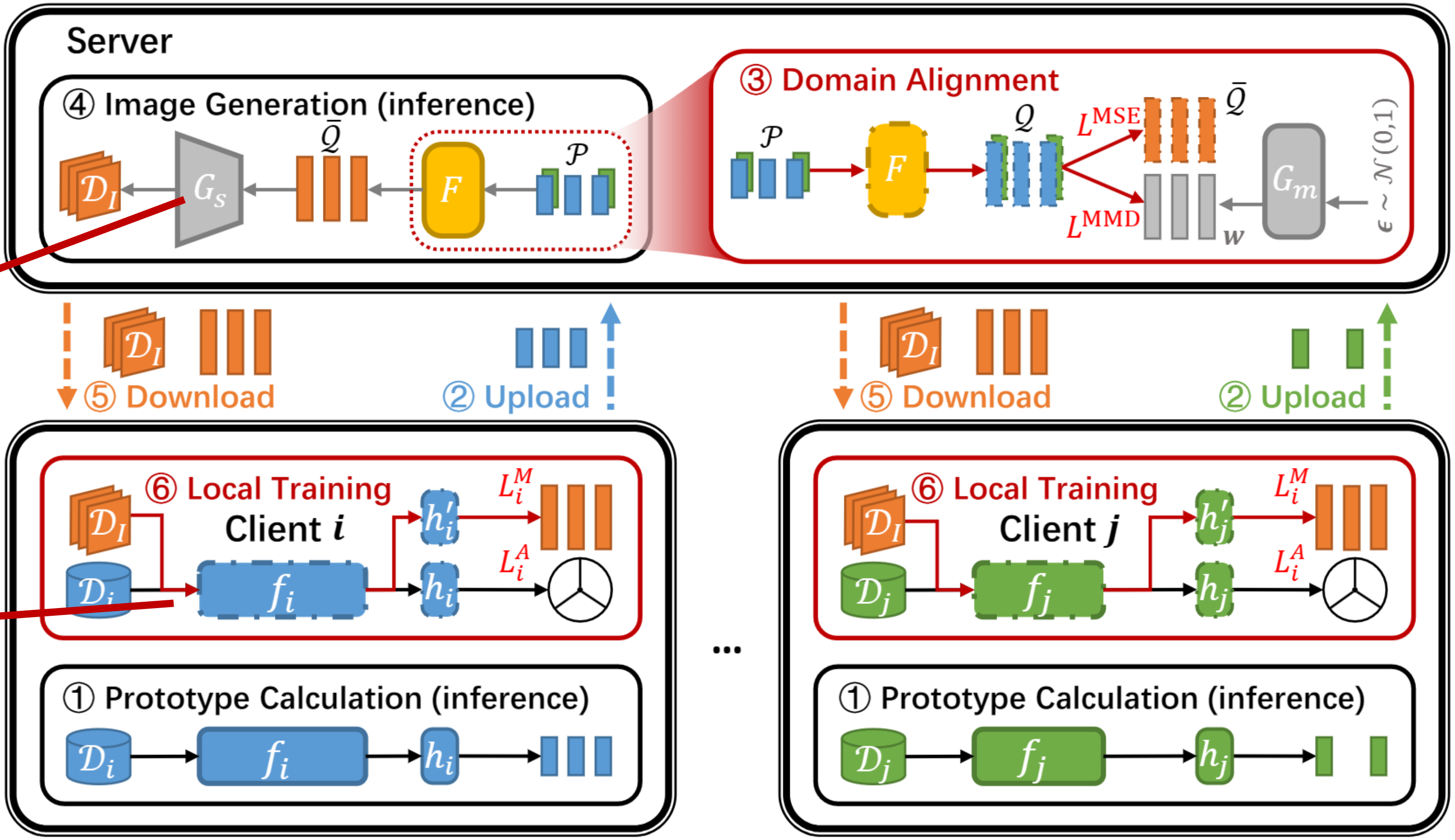
FedKTL

- Transform client prototypes to images **using a pre-trained generator on the server**
- **Transfer pre-existing knowledge** from the generator to client models



FedKTL

- Transform client prototypes to images **using a pre-trained generator on the server**
- **Transfer pre-existing knowledge** from the generator to client models



StyleGAN series
Stable Diffusion

Shallow CNN
ResNet、ViT

Accepted by CVPR 2024

Outperform baselines by up to 7.31%

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): Four images (one image per class) on client #1. (b), (c), (d), and (e): The images generated by different StyleGAN3s corresponding to the aforementioned four classes.

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 HtFE_8 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!