Summary of My Research

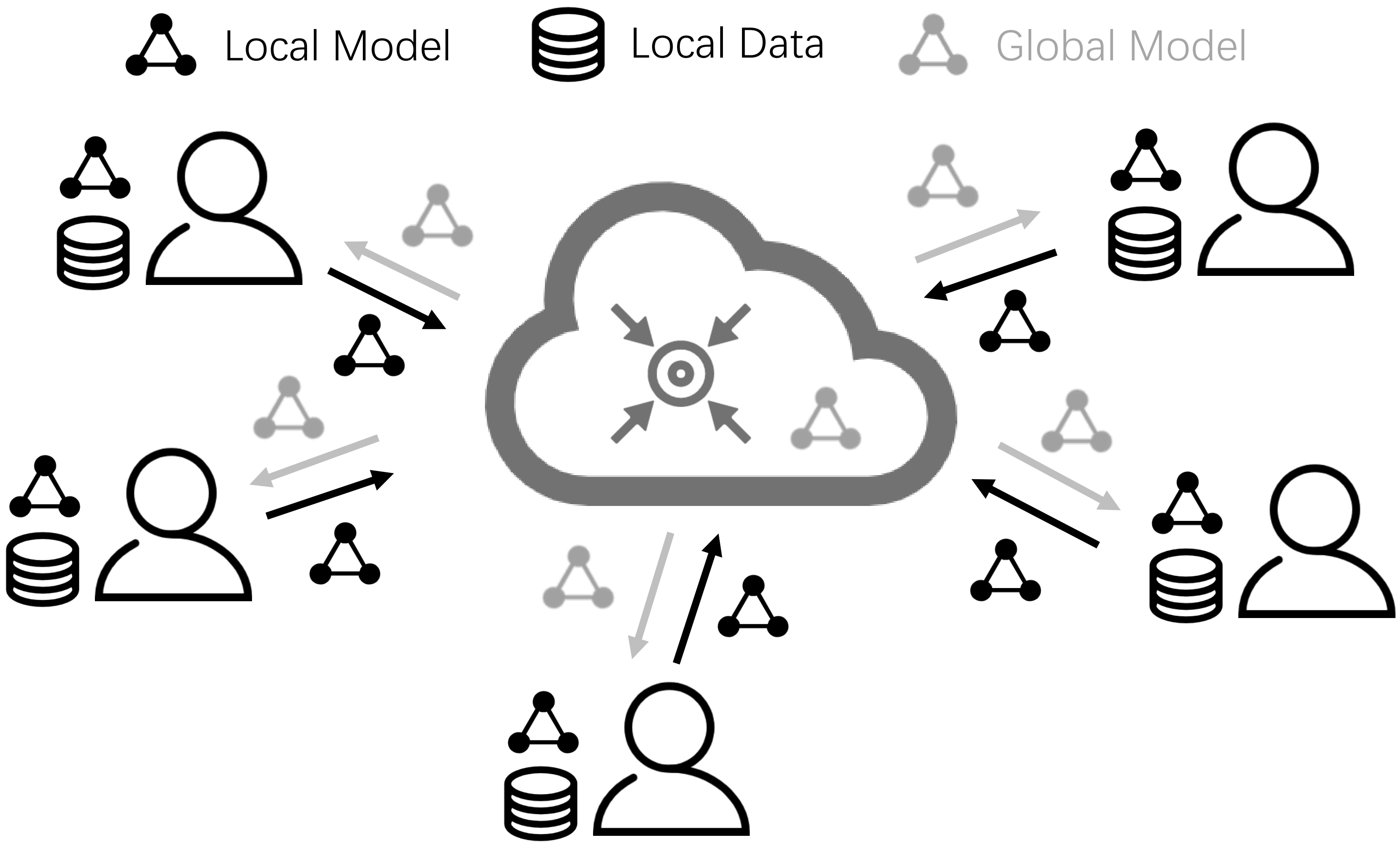
- **Name:** Jianqing Zhang
- **Age:** 27
- **Ph.D.:** Shanghai Jiao Tong University
- **Collaborations:**
  - Qiang Yang, HKUST, China
  - Yang Liu, Tsinghua University, China
  - Marco Canini, KAUST, Saudi Arabia
  - Yang Hua, Queen's University Belfast, UK

Overview

- **Research interests**
  - Personalized/Heterogeneous Federated Learning
  - Large and Small Models Collaboration
  - Synthetic Dataset Generation
- **Open-sourced projects (initiator, main contributor)**
  - CoAutoGen, PFLlib (1700+ stars, 300+ forks), HtFLlib, FL-IoT, etc.
- **Featured publications (6 accepted papers, first author)**
  - Stage ① [Personalized Federated Learning]:
    - PFLlib, AAAI'23, KDD'23, ICCV'23, NeurIPS'23, JMLR'25
  - Stage ② [Large and Heterogeneous Small Models Collaboration]:
    - HtFLlib, HtFLlib-OnDevice (private), AAAI'24, CVPR'24
  - Stage ③ [Synthetic Dataset Generation]:
    - CoAutoGen, EMNLP'24, 2x ICML'25

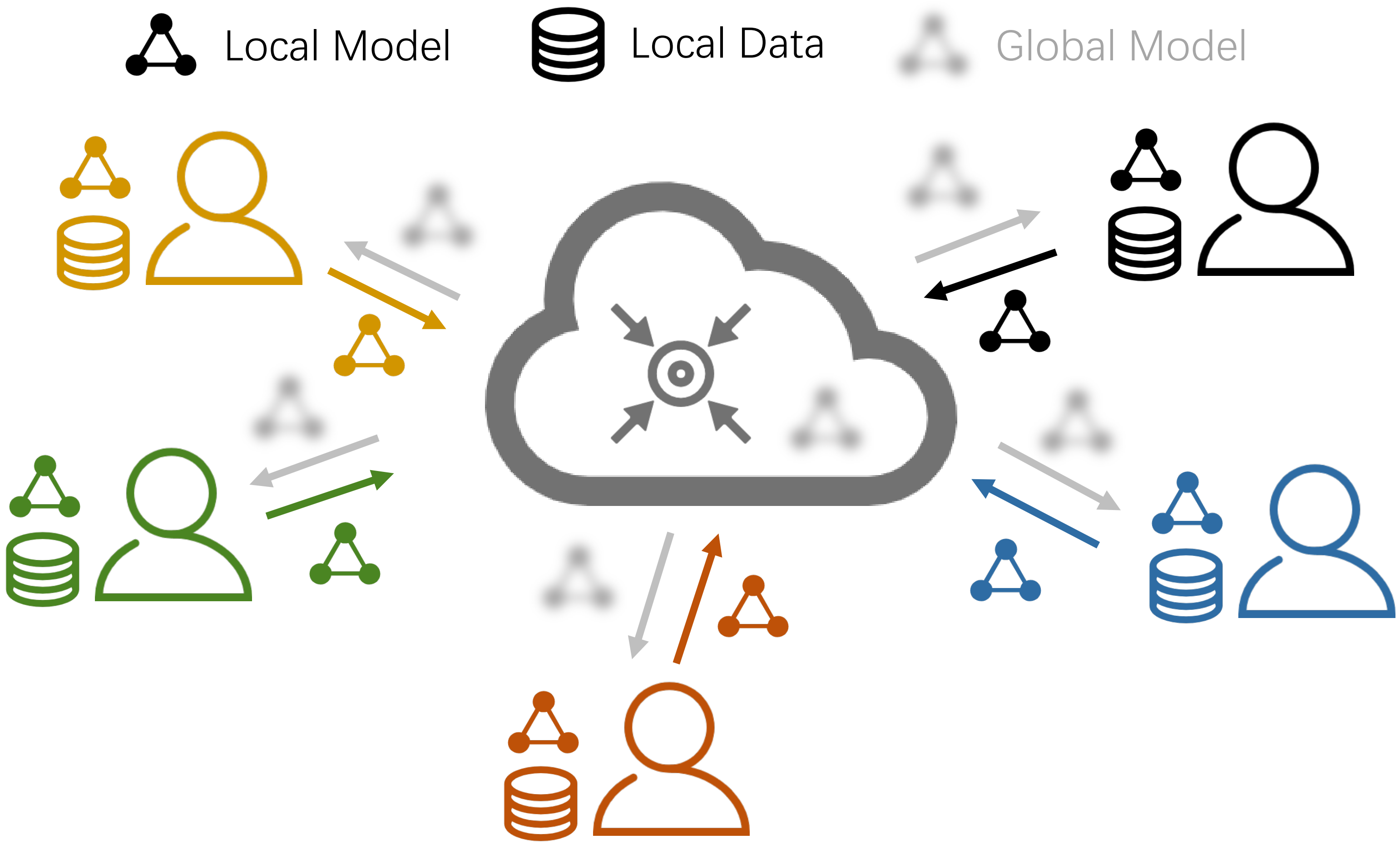
Federated Learning (FL)

- A **collaborative** and **privacy-preserving** technique for AI model training
- Finally output **one global model**



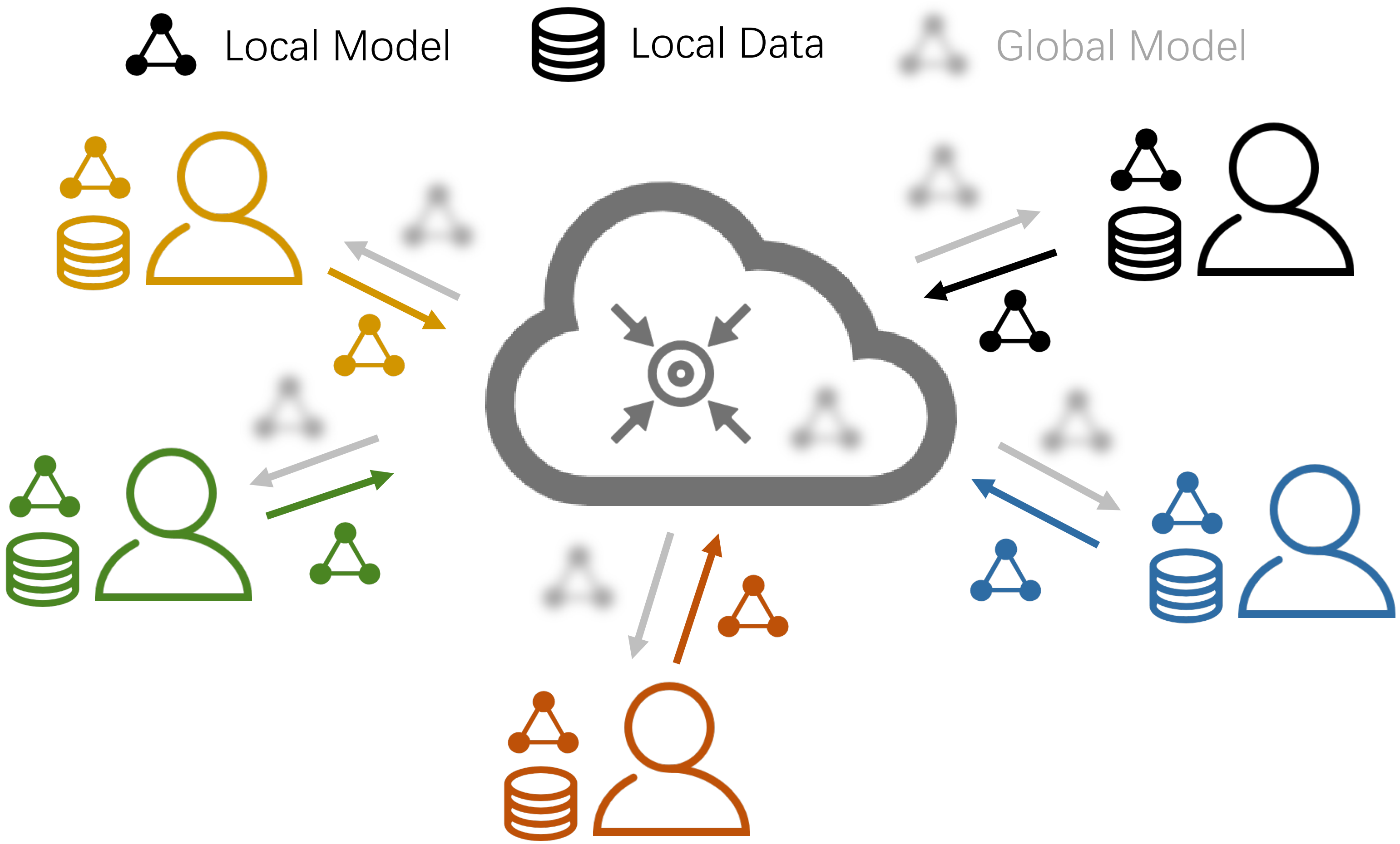
① Data Heterogeneity in FL

- Data is **generated in different ways on clients** and forbidden to be shared
- Each client also has **personalized preferences**



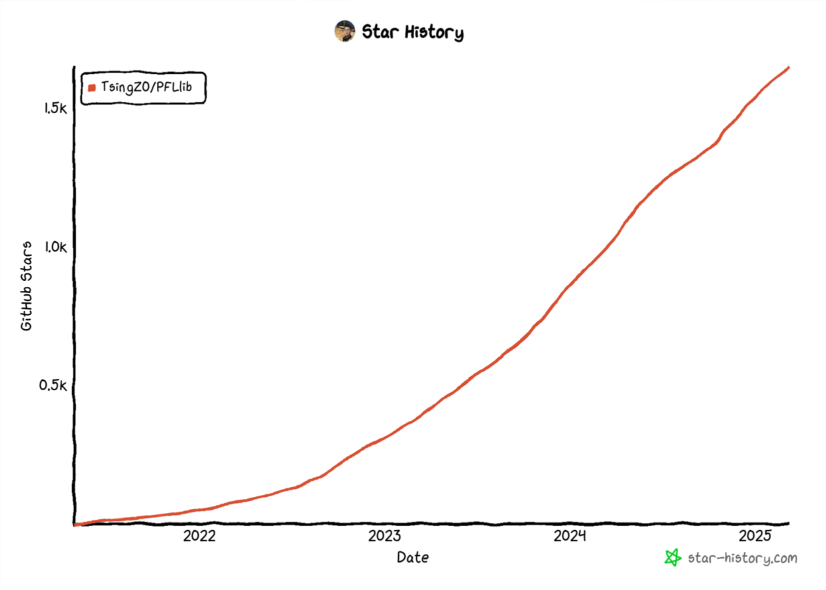
① Personalized Federated Learning (pFL)

- Utilize the **intermediate** global model to **facilitate local training**
- Finally output **personalized models**



① PFLlib: pFL algorithm library and benchmark

- **Beginner-friendly**
- **38 FL&pFL, 3 scenarios, 24 datasets**
- **Popular (1700+ stars)**
- **500 clients: 5GB GPU memory**
- **Rapidly developing:**



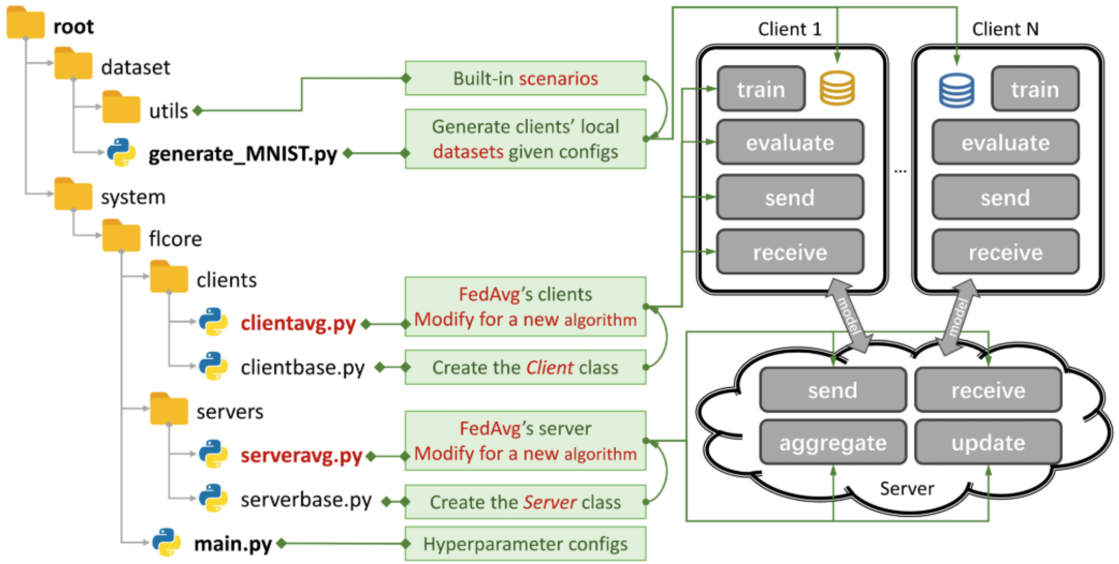
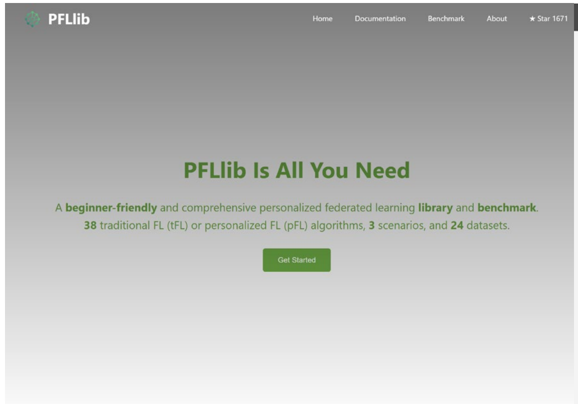
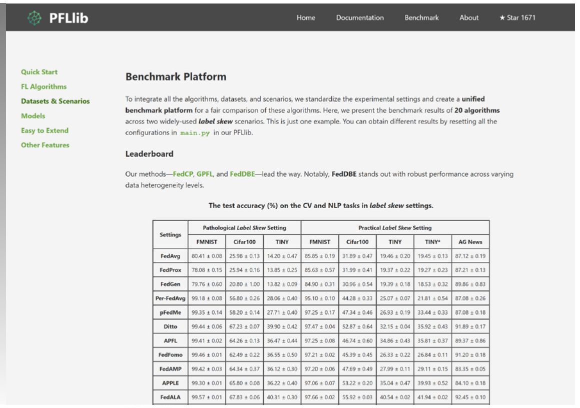


Figure 1: An Example for FedAvg. You can create a scenario using `generate_DATA.py` and run an algorithm using `main.py`, `clientNAME.py`, and `serverNAME.py`. For a new algorithm, you only need to add new features in `clientNAME.py` and `serverNAME.py`.





https://github.com/TsingZ0/PFLlib and https://www.pfllib.com/

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 distinguish both generalization and personalization?**

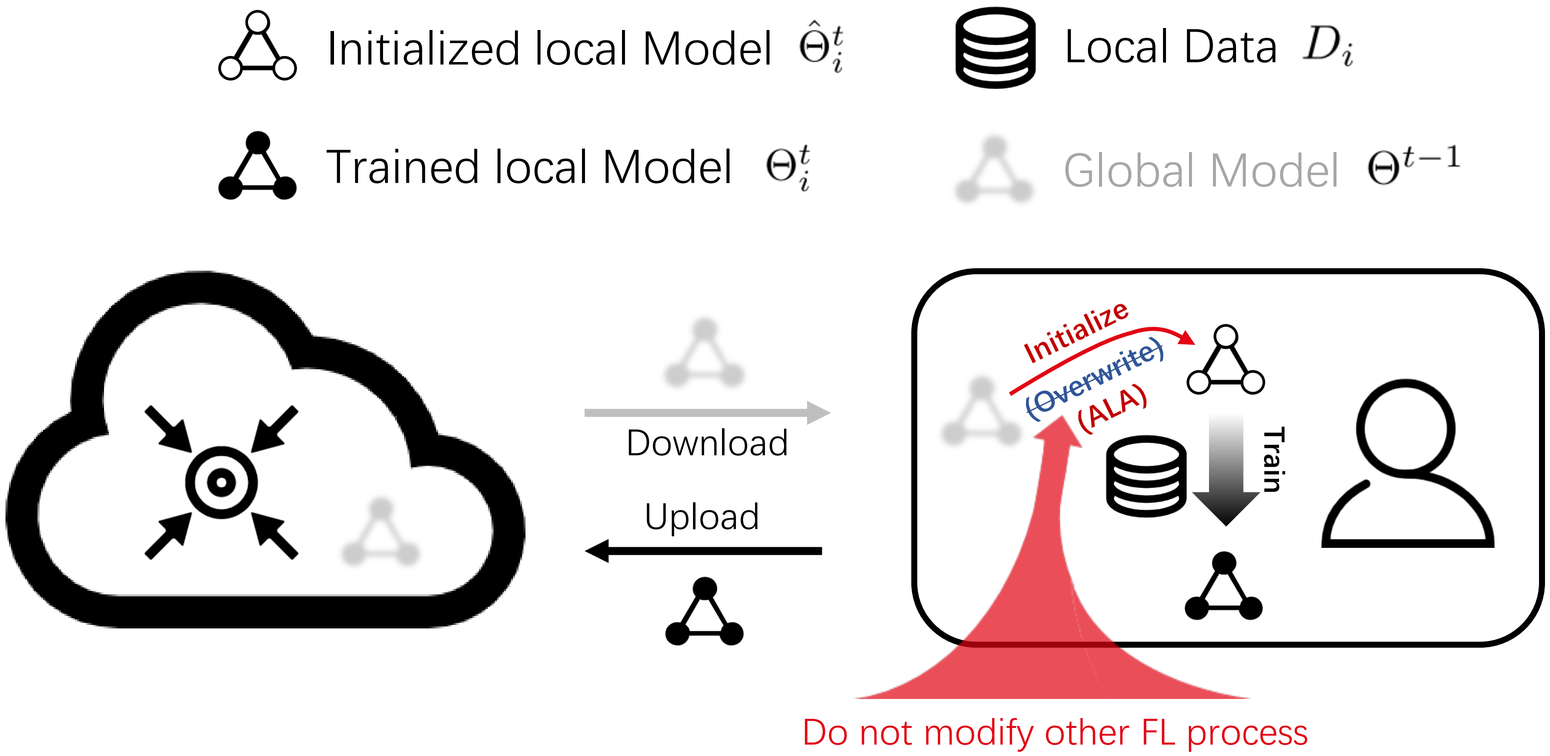
① Publications

- **[AAAI’23]** FedALA: Adaptive Local Aggregation for Personalized Federated Learning. (It has potential for pre-trained model merging[1])
- **[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.

[1] Ilharco G, Ribeiro M T, Wortsman M, et al. Editing models with task arithmetic[C]. ICLR 2023.

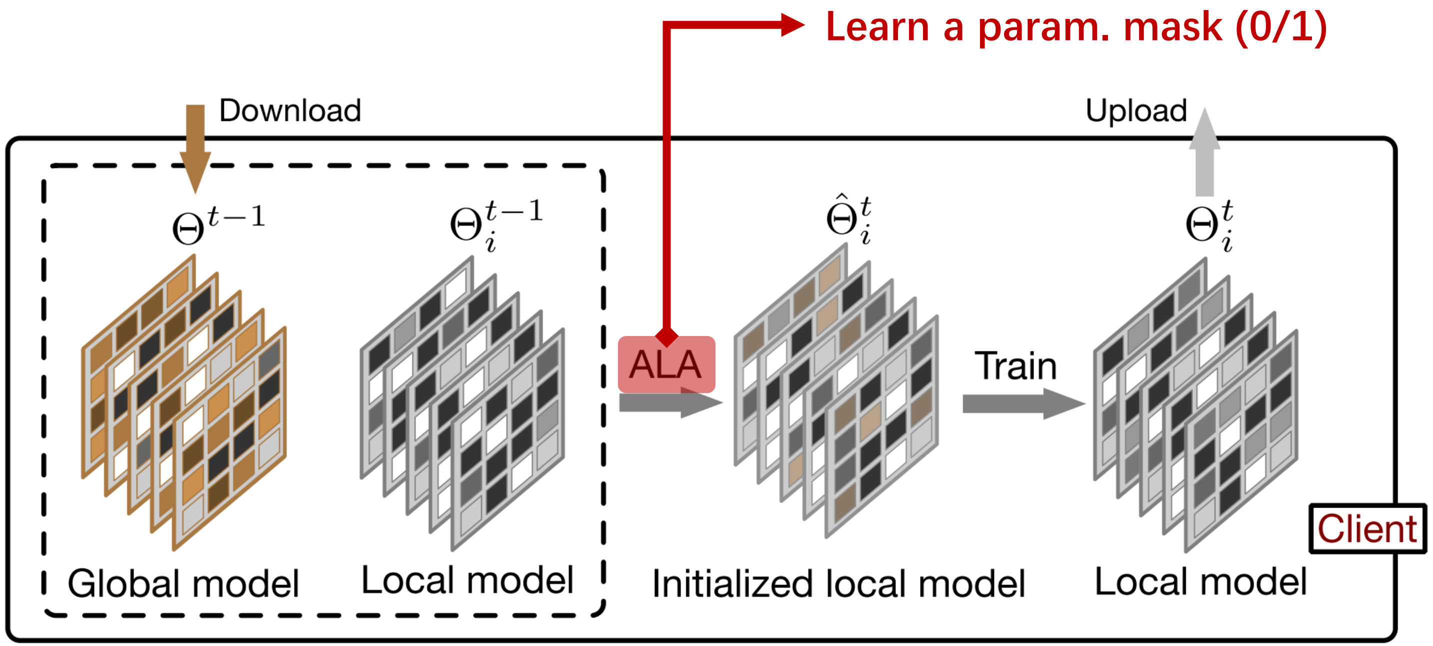
Existing FL

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



FedALA

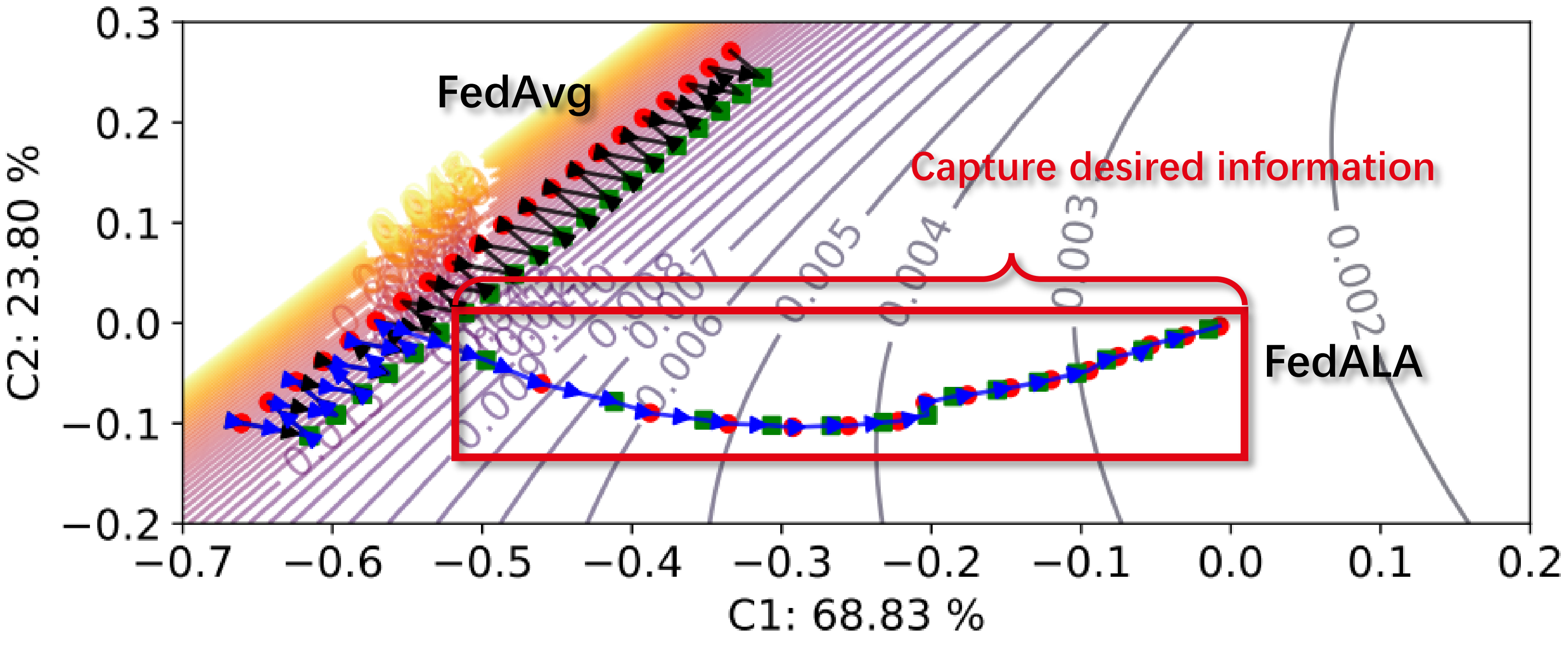
- **Extract each client's desired information from the global model** that facilitates local training
- **Adaptively aggregate** the information in the global and local model for initialization



Workflow on the client in one iteration

FedALA

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



2D visualization of local learning trajectory

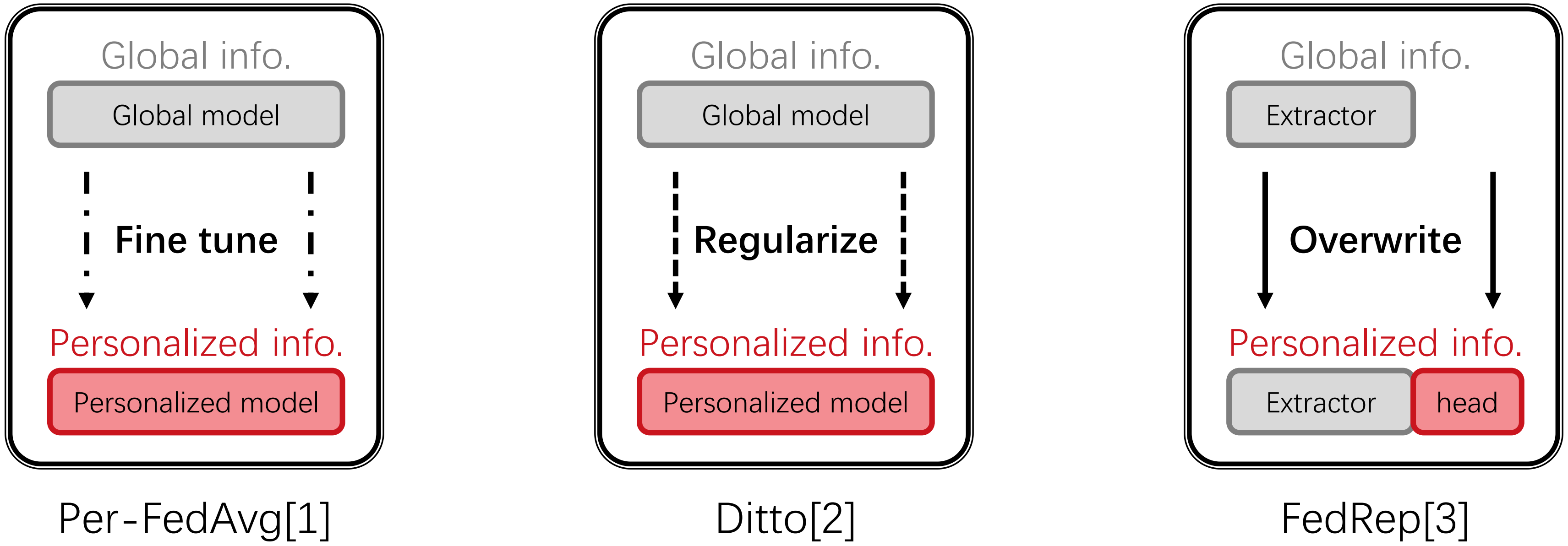
① Publications

- **[AAAI'23]** FedALA: Adaptive Local Aggregation for Personalized Federated Learning.
- **[KDD'23]** FedCP: Separating Feature Information for Personalized Federated Learning via Conditional Policy. (It has potential for multifunctional foundation model merging[1])
- **[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.

[1] Ranzinger M, Heinrich G, Kautz J, et al. AM-RADIO: Agglomerative Vision Foundation Model Reduce All Domains Into One[C]. CVPR 2024.

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.

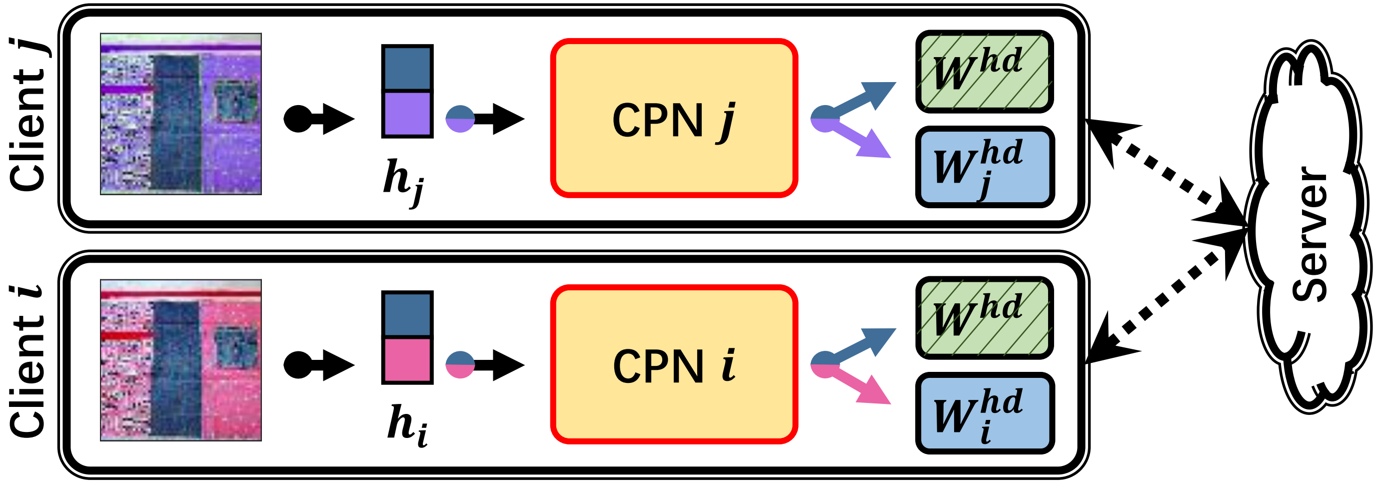


- They only focus on model parameters, but **ignore the source of information: data.**

[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

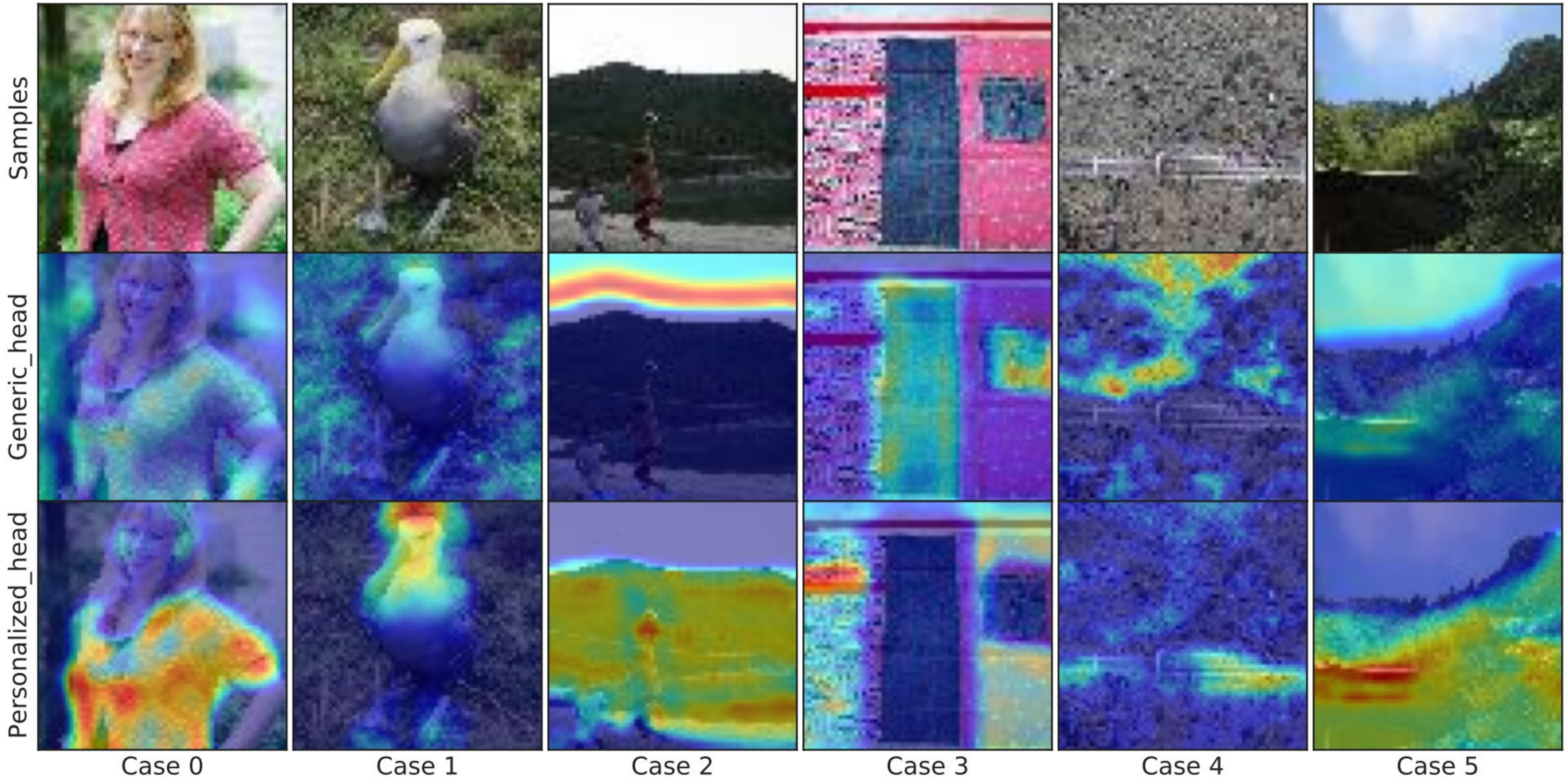
- We separate feature information via an auxiliary **Conditional Policy Network (CPN).**
  - Sample-specific separation
  - Lightweight (e.g., 4.67% parameters of ResNet-18)



- Then, we utilize global and personalized information via global and personalized heads.

FedCP

- **Separating Feature Information**



Six samples from the Tiny-ImageNet dataset

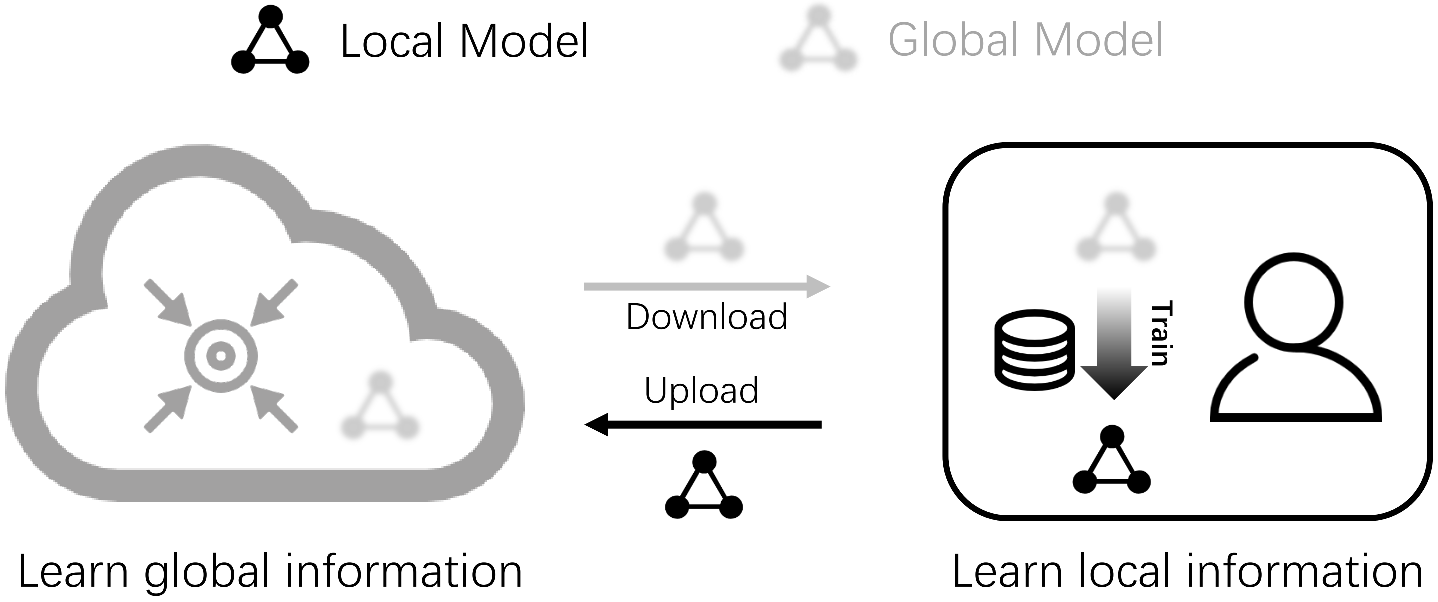
① Publications

- **[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. (It has potential for multifunctional foundation model merging[1])
- **[NeurIPS'23]** Eliminating Domain Bias for Federated Learning in Representation Space.

[1] Wang H, Vasu P K A, Faghri F, et al. Sam-clip: Merging vision foundation models towards semantic and spatial understanding[C].CVPR 2024.

Existing pFL

- Most pFL learns global and personalized information **alternatively**
- **Catastrophic forgetting**: global knowledge is lost during local training



GPFL

- GCE introduces more global information **simultaneously** with local training
- CoV **eliminates the interaction** between global and personalized feature learning

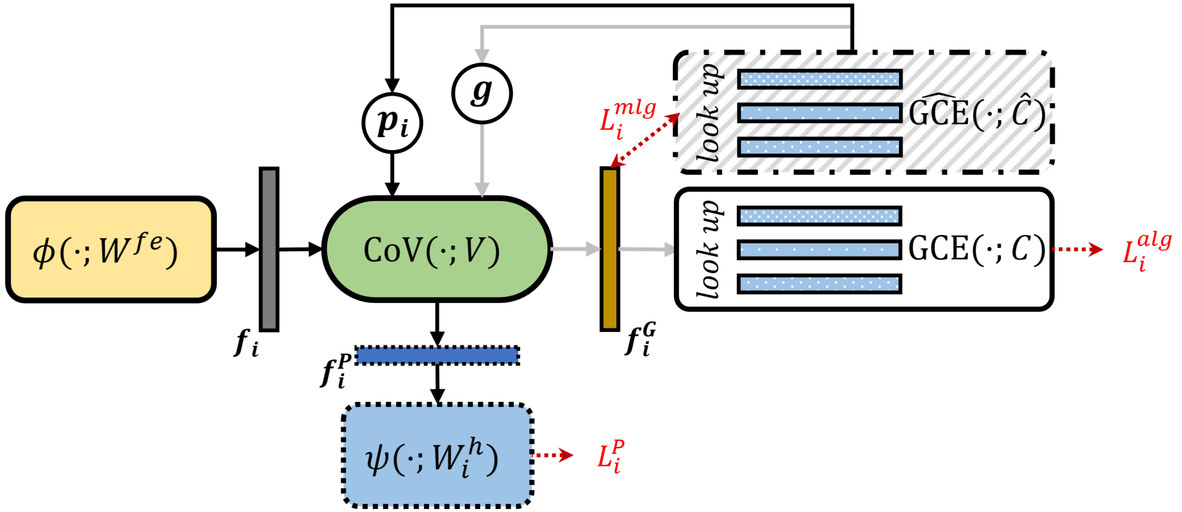
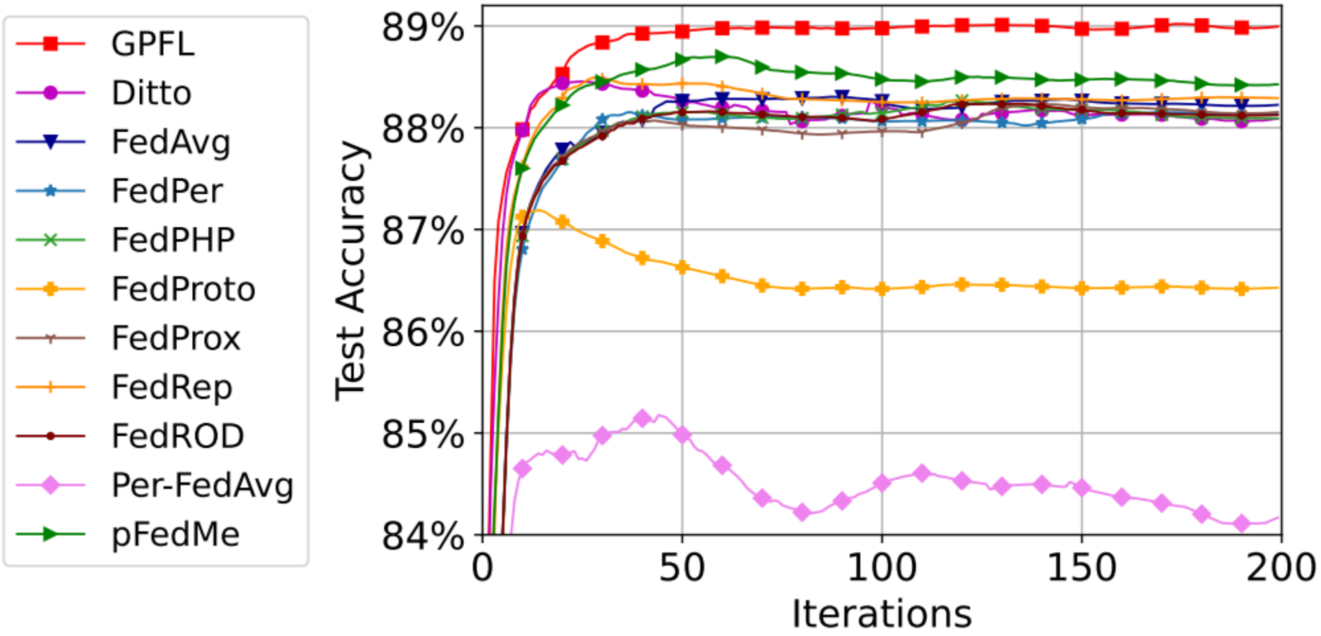


Illustration of client modules and data flow between them

GPFL

- Relieve the **widely existed** overfitting issue in pFL



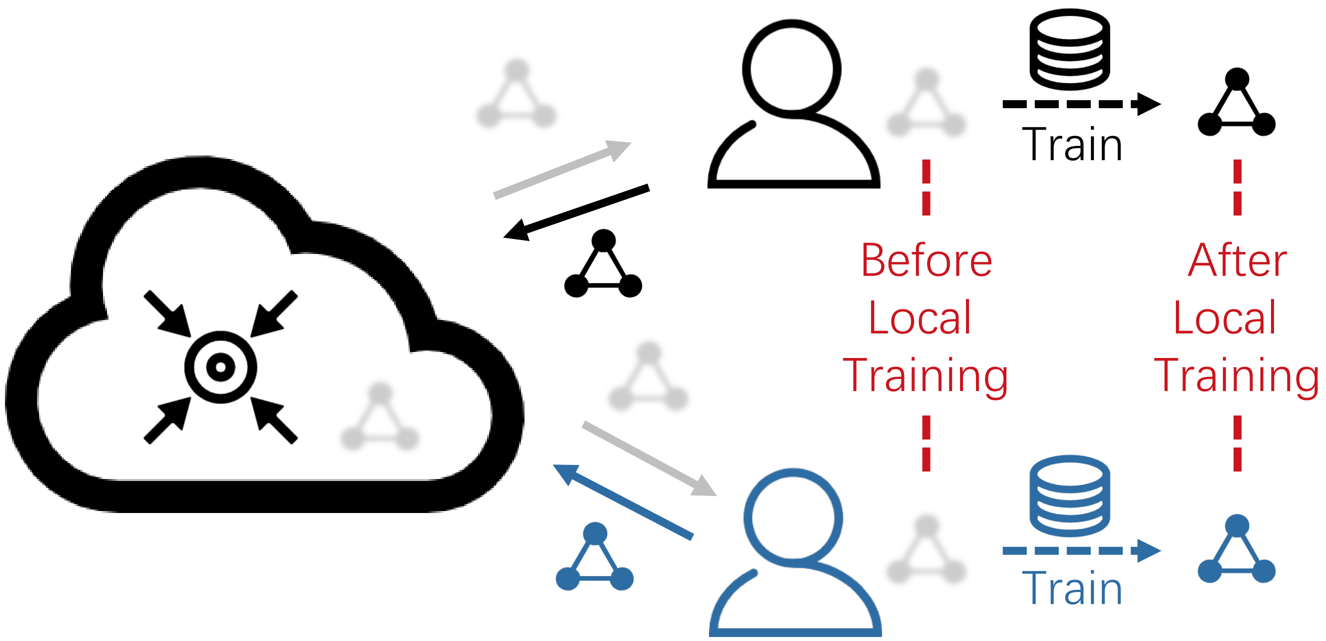
Test accuracy curves in the feature shift setting

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.

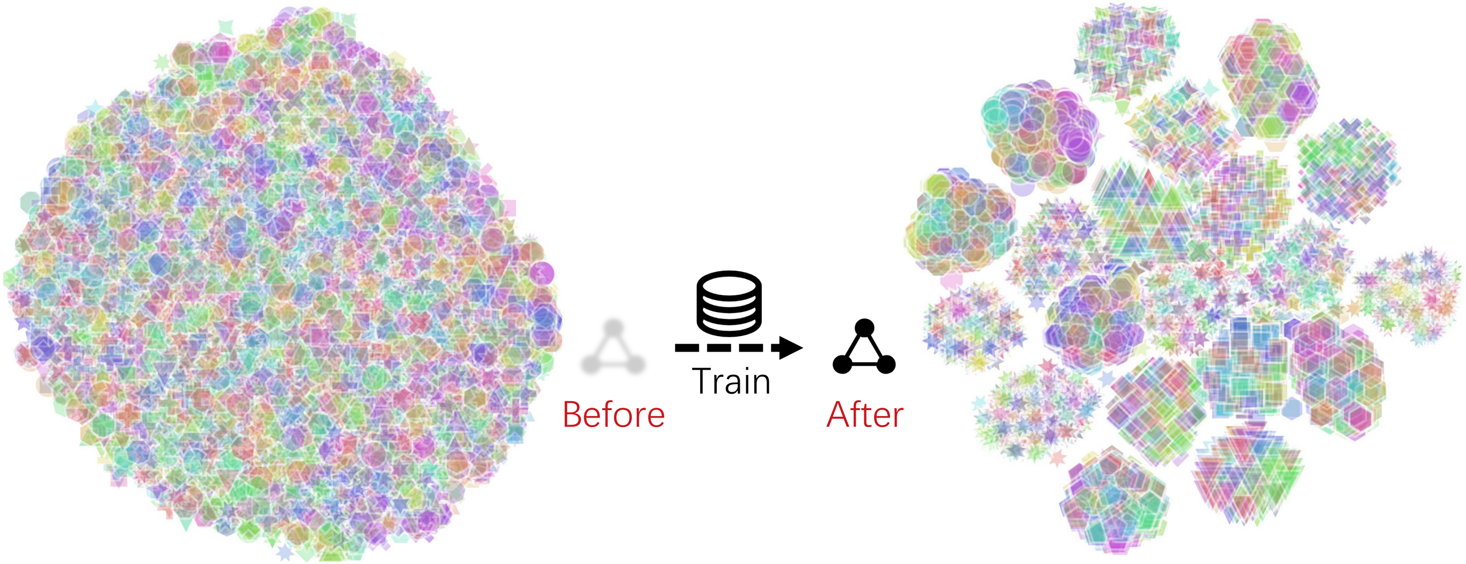
Existing FL

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



Existing FL

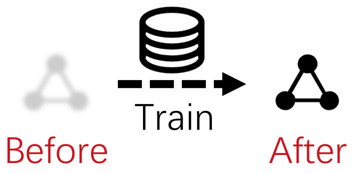
- After local training, the feature representations are **biased** to client-specific domains



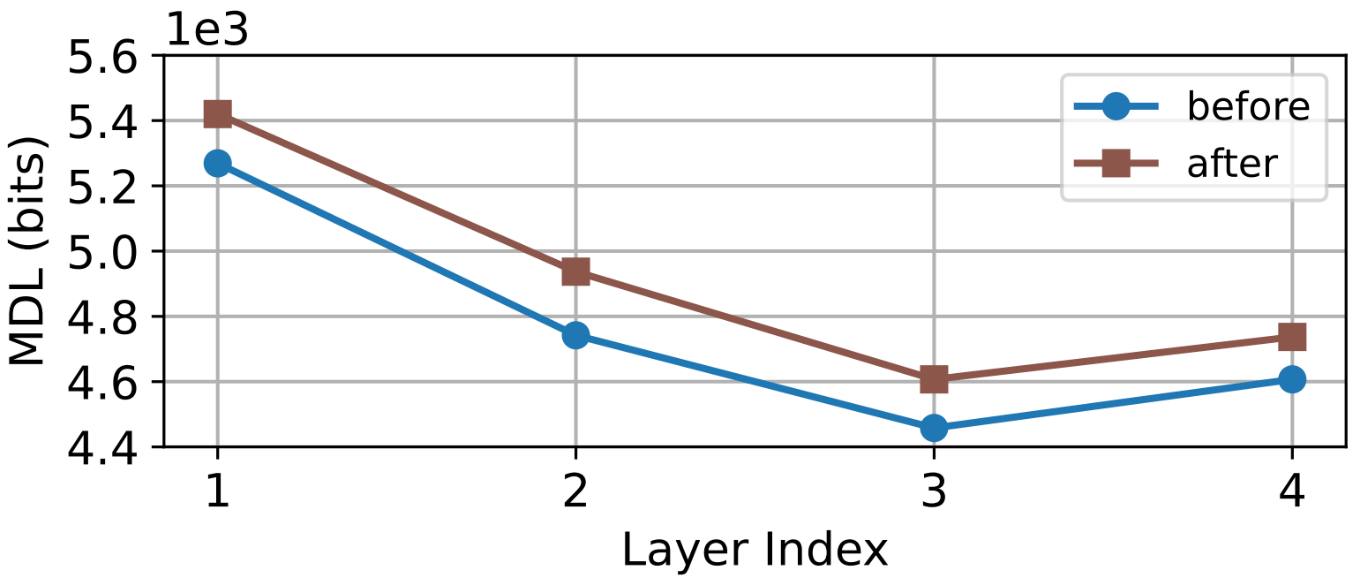
(a) Before local training (b) After local training

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

Existing FL



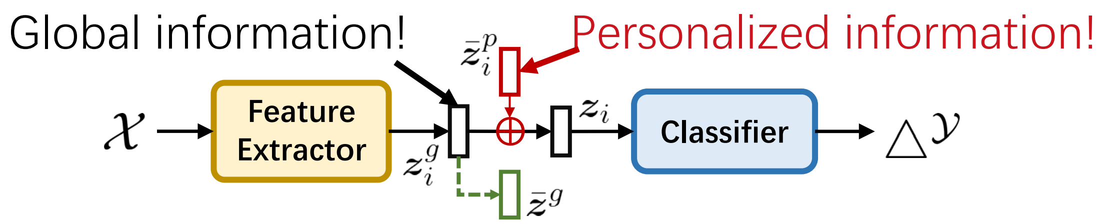
- At the same time, representations' quality is also **degenerated**



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

DBE

- Eliminate domain bias by store **personalized information** in **PRBM**
- Enhance **information disentanglement** by guiding feature extractor with **MR**



Local model (with PRBM and MR)

DBE

- Improve **bi-directional knowledge transfer**
- **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

- Improve **bi-directional knowledge transfer**
- **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* $\texttt{PRBM} : \mathcal{Z} \mapsto \mathcal{Z}$ *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^*, $\mathcal{F}' = \texttt{PRBM} \circ \mathcal{F}^g : \mathcal{X} \mapsto \mathcal{Z}$, then for every $h' \in \mathcal{H}$, with probability at least $1 - \delta$:*

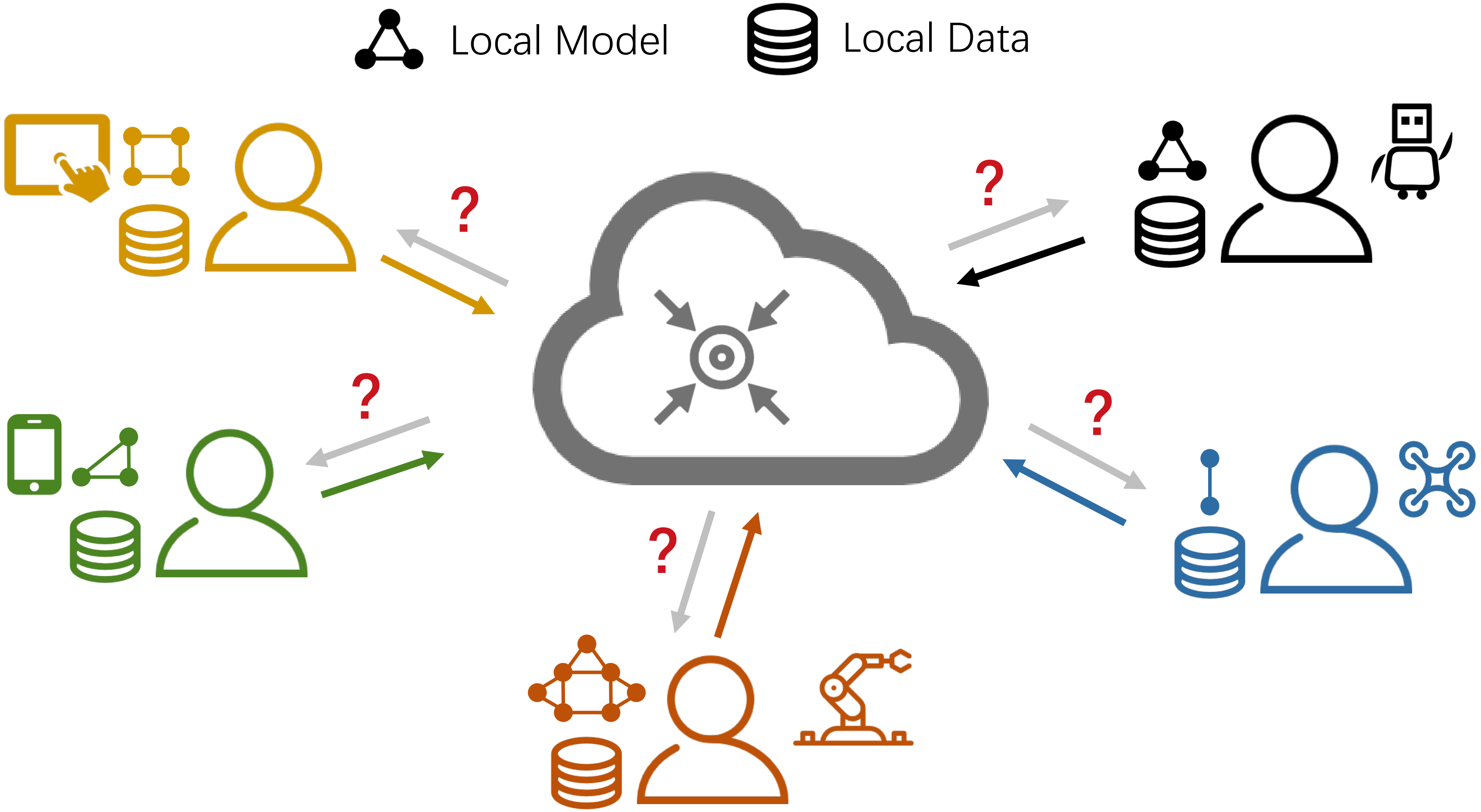
$$\mathcal{L}_{\mathcal{D}_i}(h') \leq \mathcal{L}_{\hat{\mathcal{D}}}(h') + \sqrt{\frac{4}{m}\left(d \log \frac{2em}{d} + \log \frac{4}{\delta}\right)} + \boxed{d_{\mathcal{H}}(\tilde{\mathcal{U}}', \tilde{\mathcal{U}}_i')} + \lambda_i,$$

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.

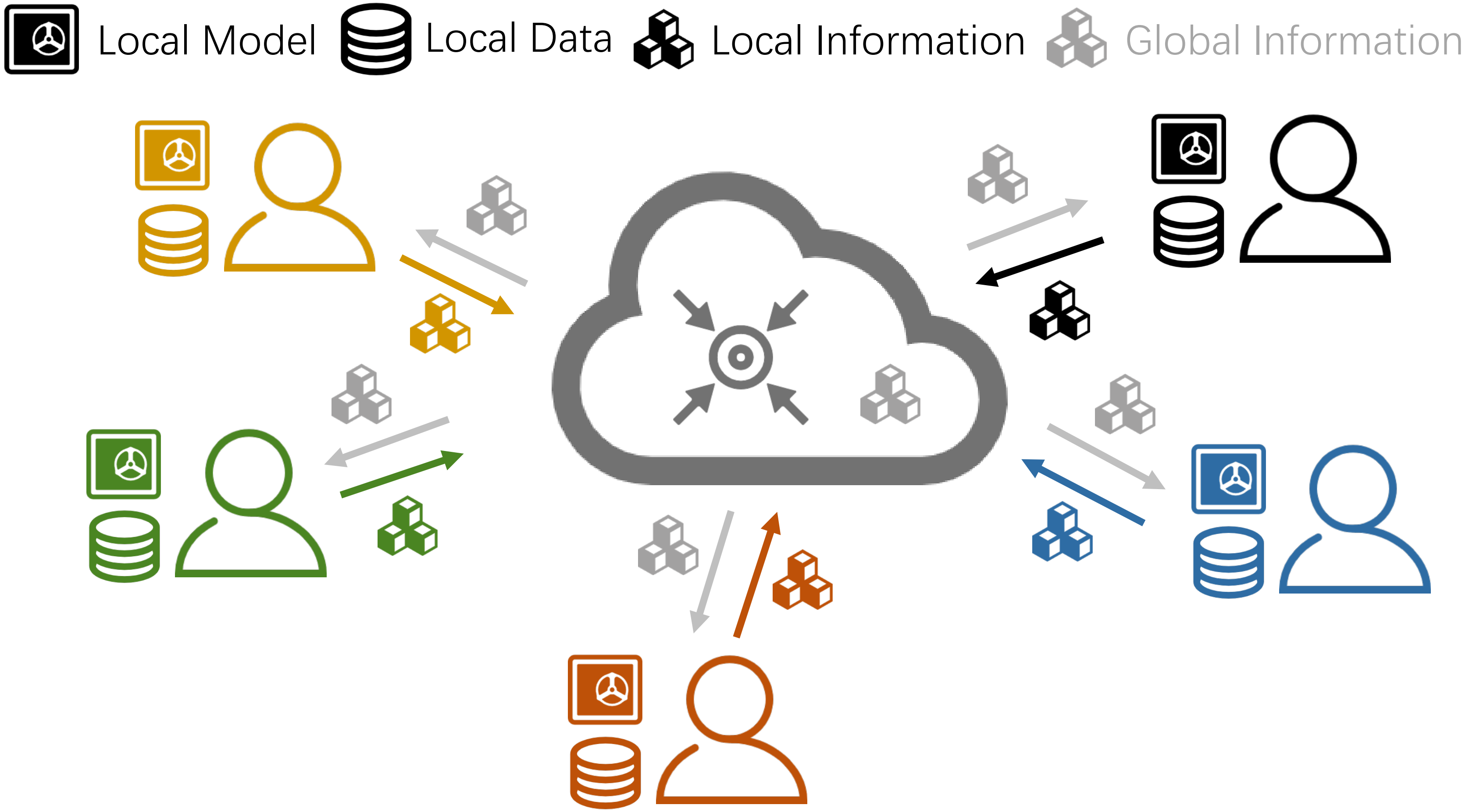
② Data and Model Heterogeneity in FL

- **Device heterogeneity** and **intellectual property**
- **Low bandwidth**: What should be transmitted?



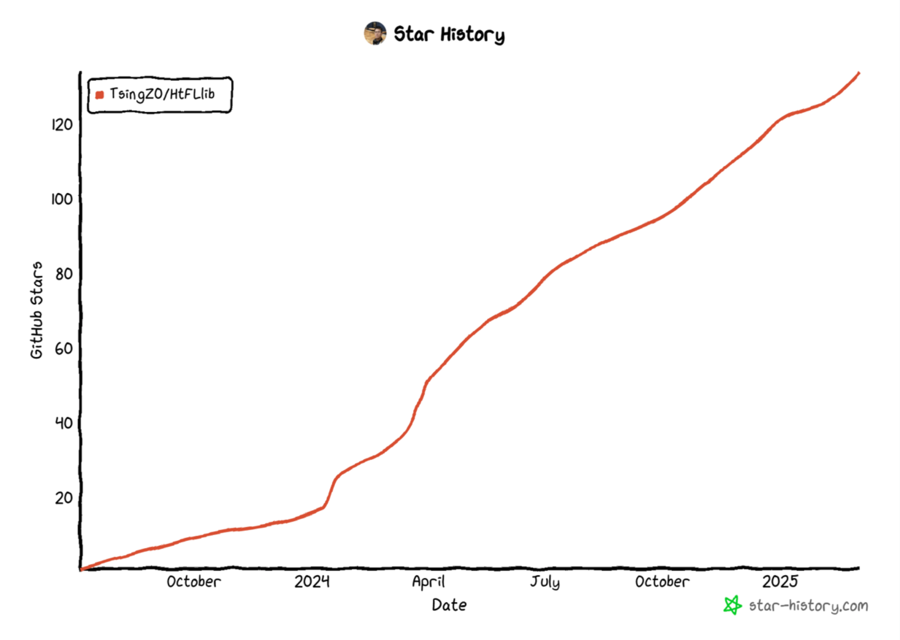
② [Large and Heterogeneous Small Models Collaboration]

- Transmits **lightweight information carriers** instead of exposing model parameters
- Typically uses **knowledge distillation**-based approaches



② HtFLlib: heterogeneous FL algorithm library

- **Beginner-friendly**
- **PFLlib compatible**
- **10 data-free HtFL**
- **40 heterogeneous models**
- **Main contributor**



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)

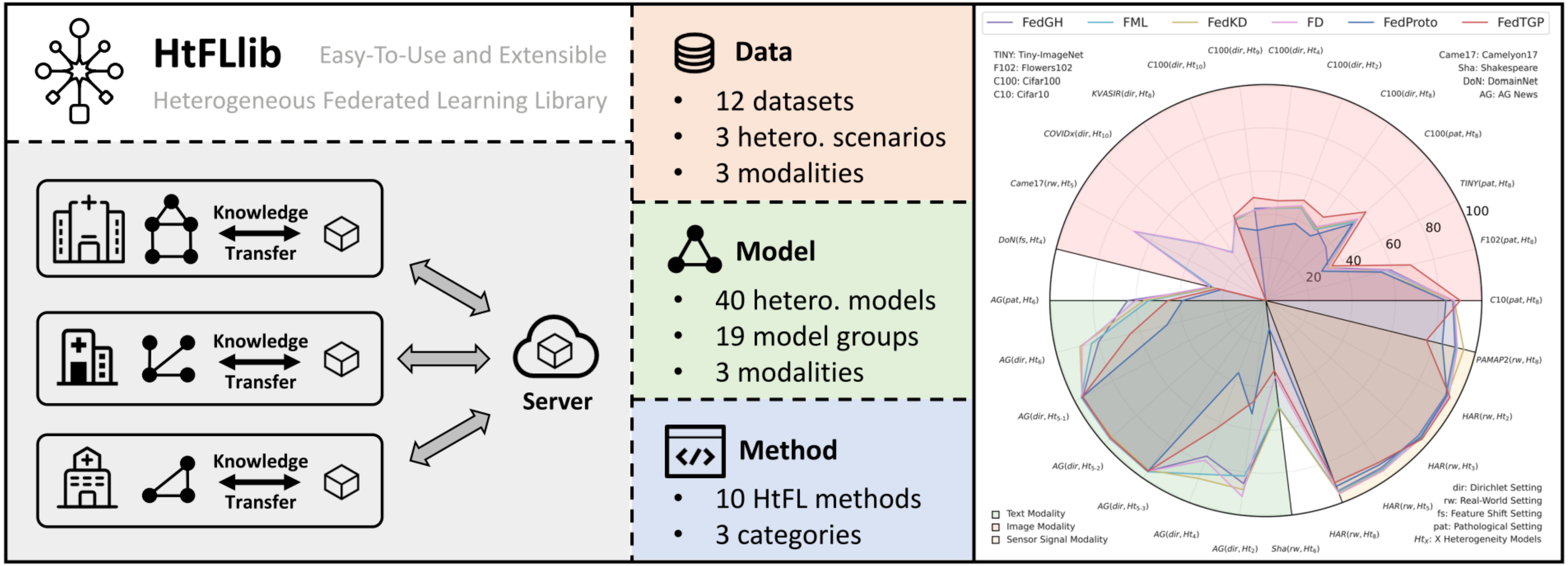
Here, "data-free" refers to the absence of any additional dataset beyond the clients' private data. We only consider data-free algorithms here, as they have fewer restrictions and assumptions, making them more valuable and easily extendable to other scenarios, such as the existence of public server data.

- **Local** — Each client trains its model locally without federation.
- **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.)*
- **FedMRL** — Federated Model Heterogeneous Matryoshka Representation Learning *NeurIPS 2024*

https://github.com/TsingZ0/HtFLlib

② HtFLlib: heterogeneous FL algorithm library

- **Beginner-friendly**
- **PFLlib compatible**
- **Extensible**



https://github.com/TsingZ0/HtFLlib

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.
  - The first one that integrates a large generative model (Stable Diffusion, etc.) into heterogeneous small models (ResNets, ViTs, etc.) collaboration

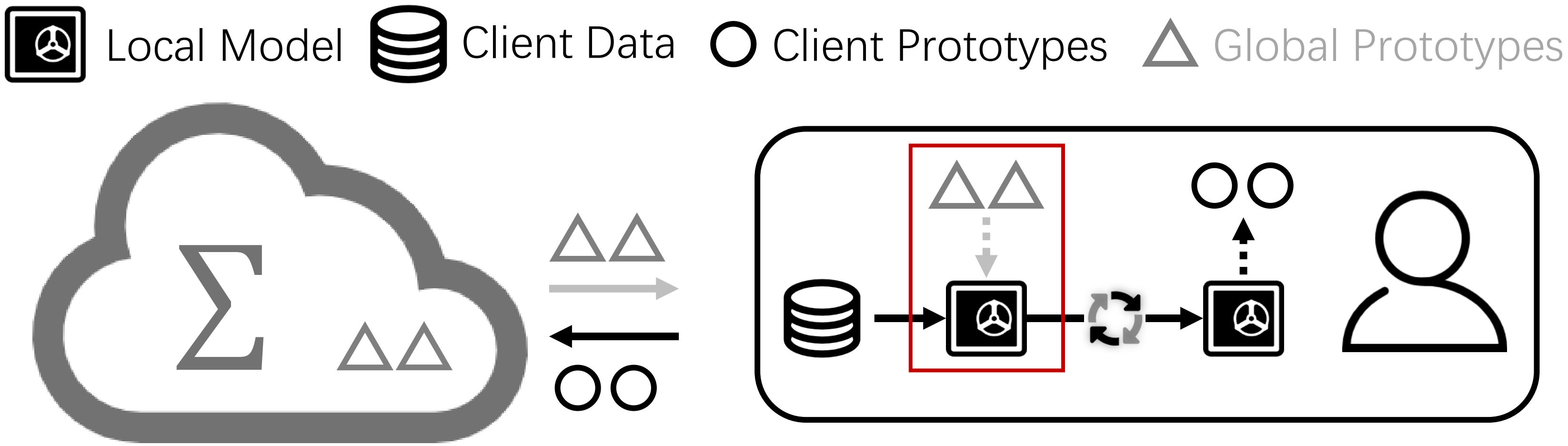
- **How can knowledge be shared and aggregated to benefit participants?**

② Publications

- **[AAAI'24]** FedTGP: Trainable Global Prototypes with Adaptive-Margin-Enhanced Contrastive Learning for Data and Model Heterogeneity in Federated Learning.
- **[CVPR'24]** An Upload-Efficient Scheme for Transferring Knowledge From a Server-Side Pre-trained Generator to Clients in Heterogeneous Federated Learning.
  - The first one that integrates a large generative model (Stable Diffusion, etc.) into heterogeneous small models (ResNets, ViTs, etc.) collaboration

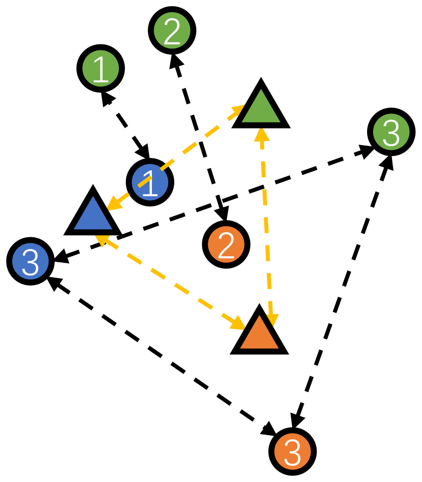
Existing HtFL

- Share **client prototypes** (class representatives) with the server
- Aggregate client prototypes to generate **global prototypes**
- Train client models in a **knowledge distillation** manner in feature space

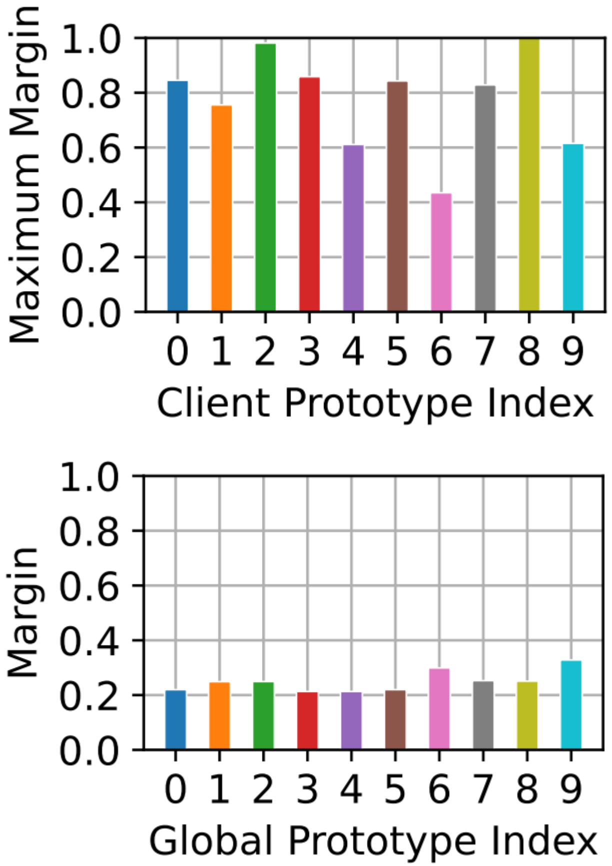


Existing HtFL

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

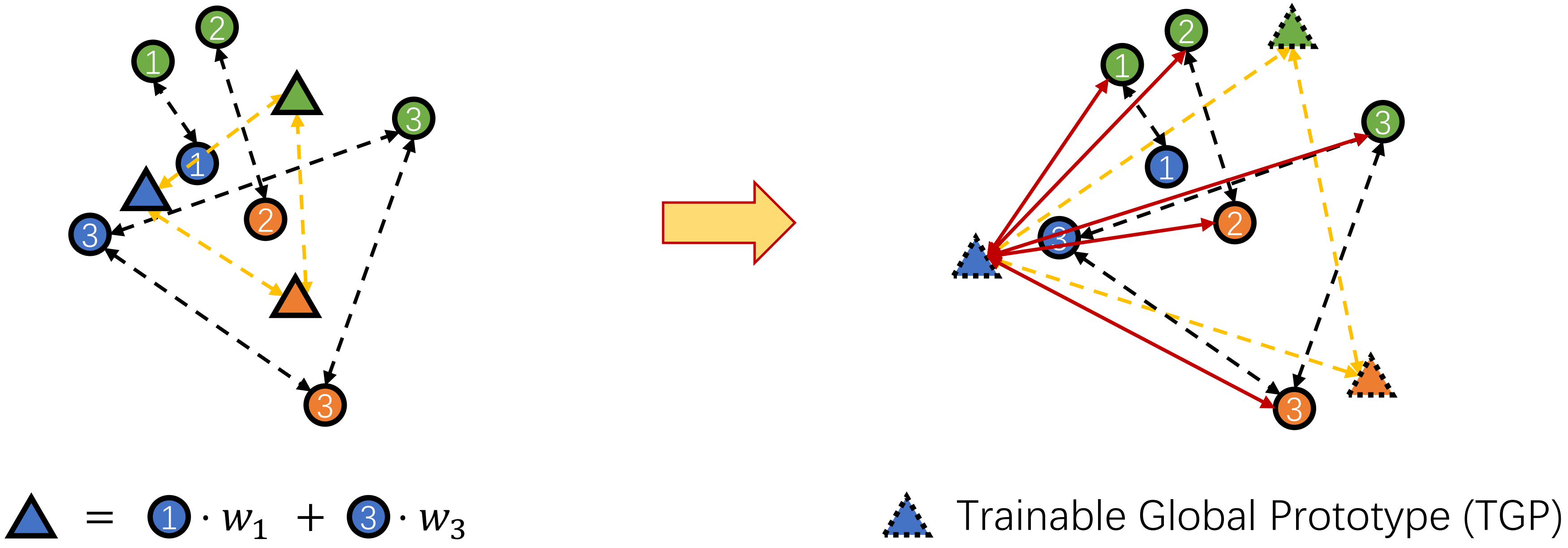


$$\triangle = ① \cdot w_1 + ③ \cdot w_3$$



FedTGP

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

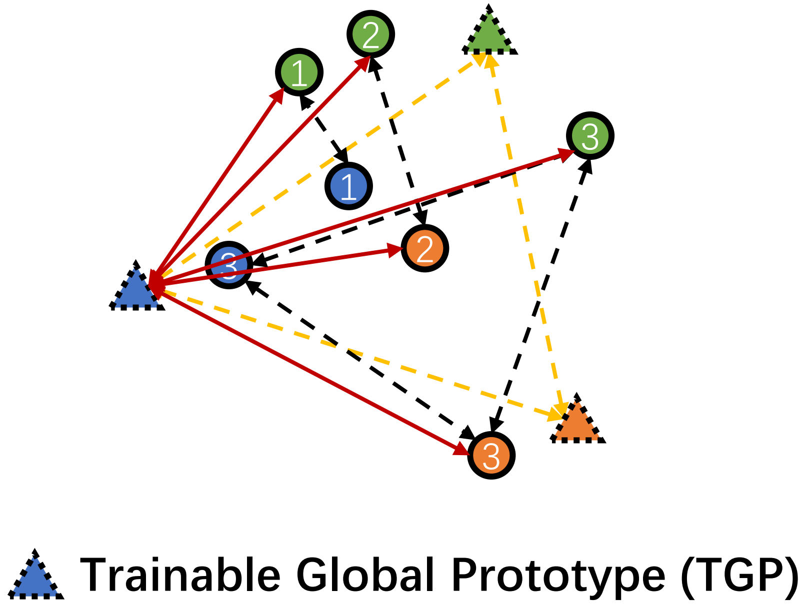


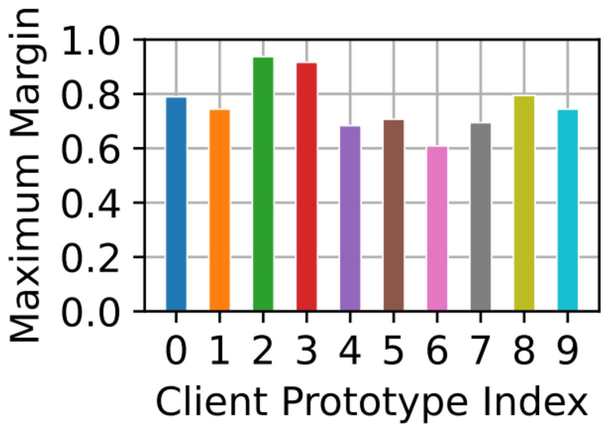
Accepted by AAAI 2024

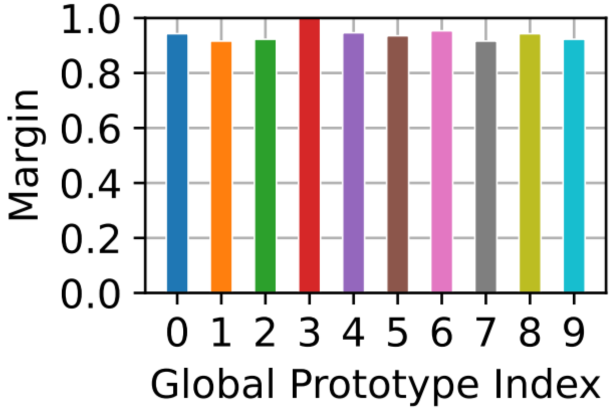
Outperform baselines by up to 9.08%, and achieve up to an 18.96% improvement over FedProto

FedTGP

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







Outperform baselines by up to 9.08%, and achieve up to an 18.96% improvement over FedProto

FedTGP

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

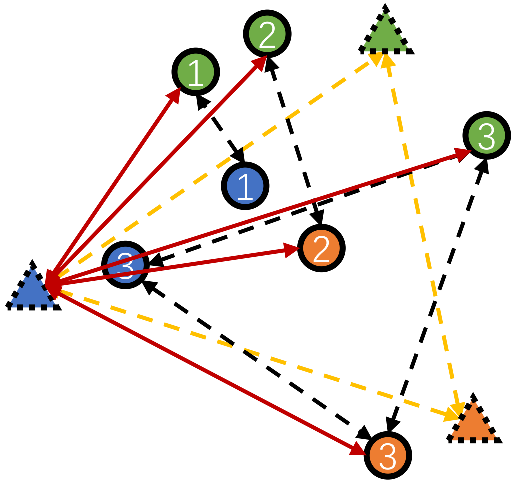
$$\min_{\hat{\mathcal{P}}} \sum_{c=1}^{C} \mathcal{L}_P^c,$$

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

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

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

Accepted by AAAI 2024

Outperform baselines by up to 9.08%, and achieve up to an 18.96% improvement over FedProto

FedTGP

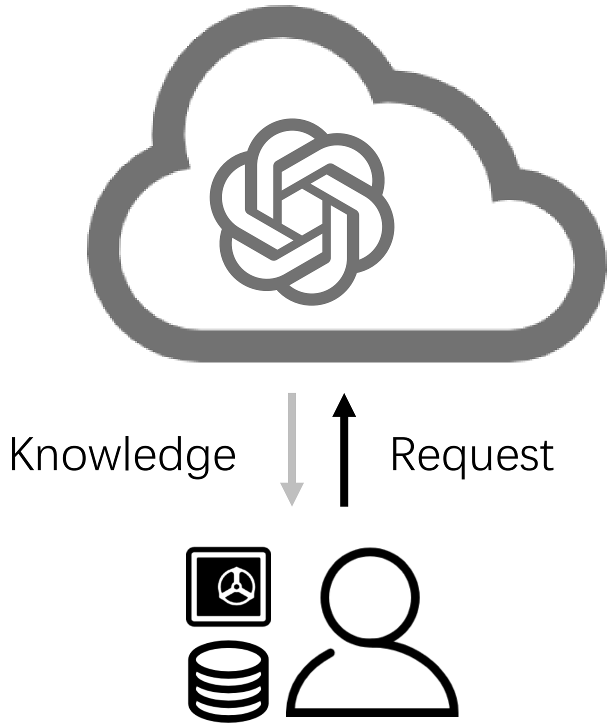
- **ACL** can also be applied to other tasks and scenarios

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

② Data Scarcity Issue

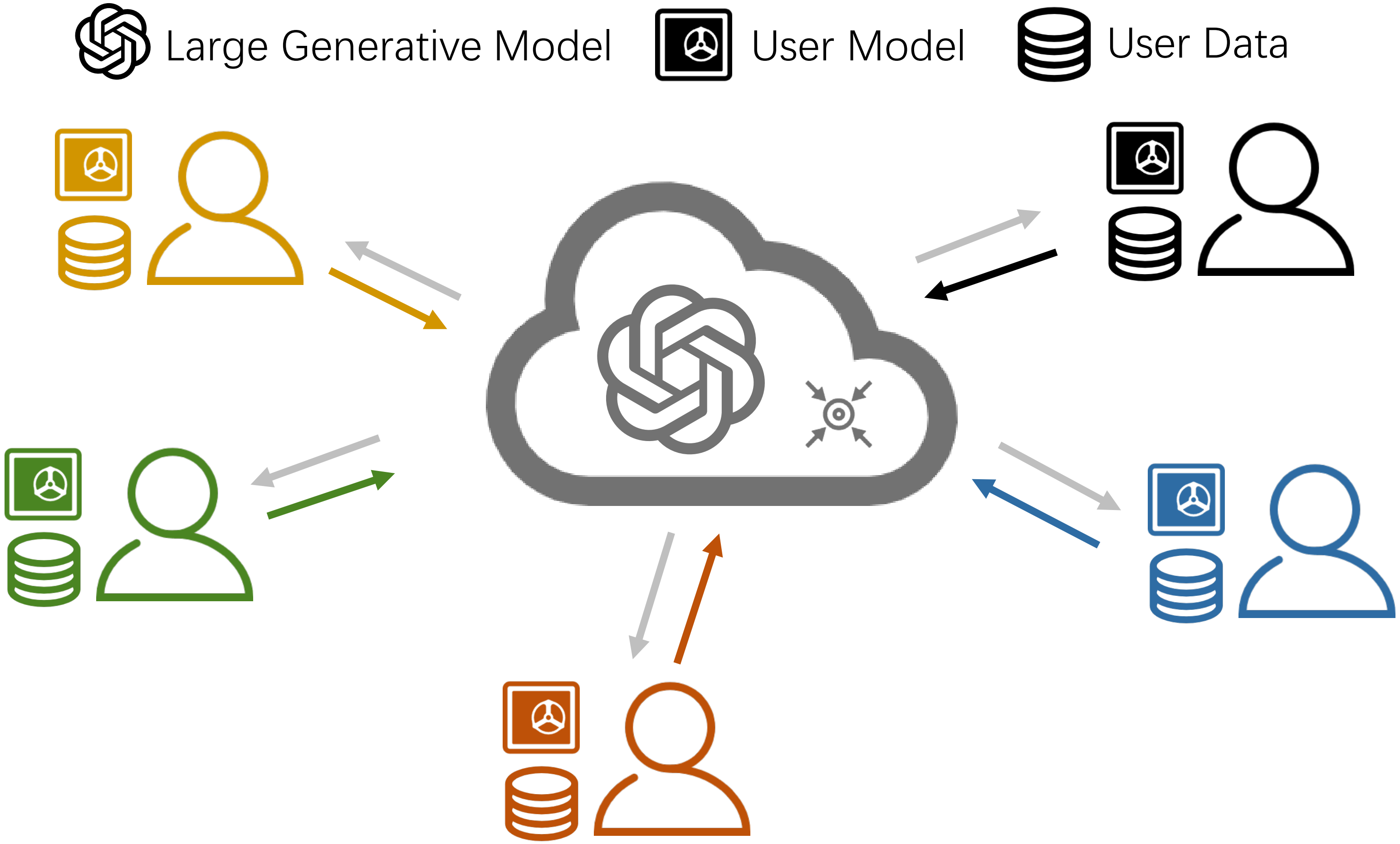
- Specific domains (e.g., **medical domain**) suffer from **data scarcity** and **privacy**
- **Generated data** serves as additional **common knowledge**

Large Generative Model User Model User Data



② [Large and Heterogeneous Small Models Collaboration]

- Transfer **common knowledge** from server to clients
- Transfer **task-specific knowledge** among clients

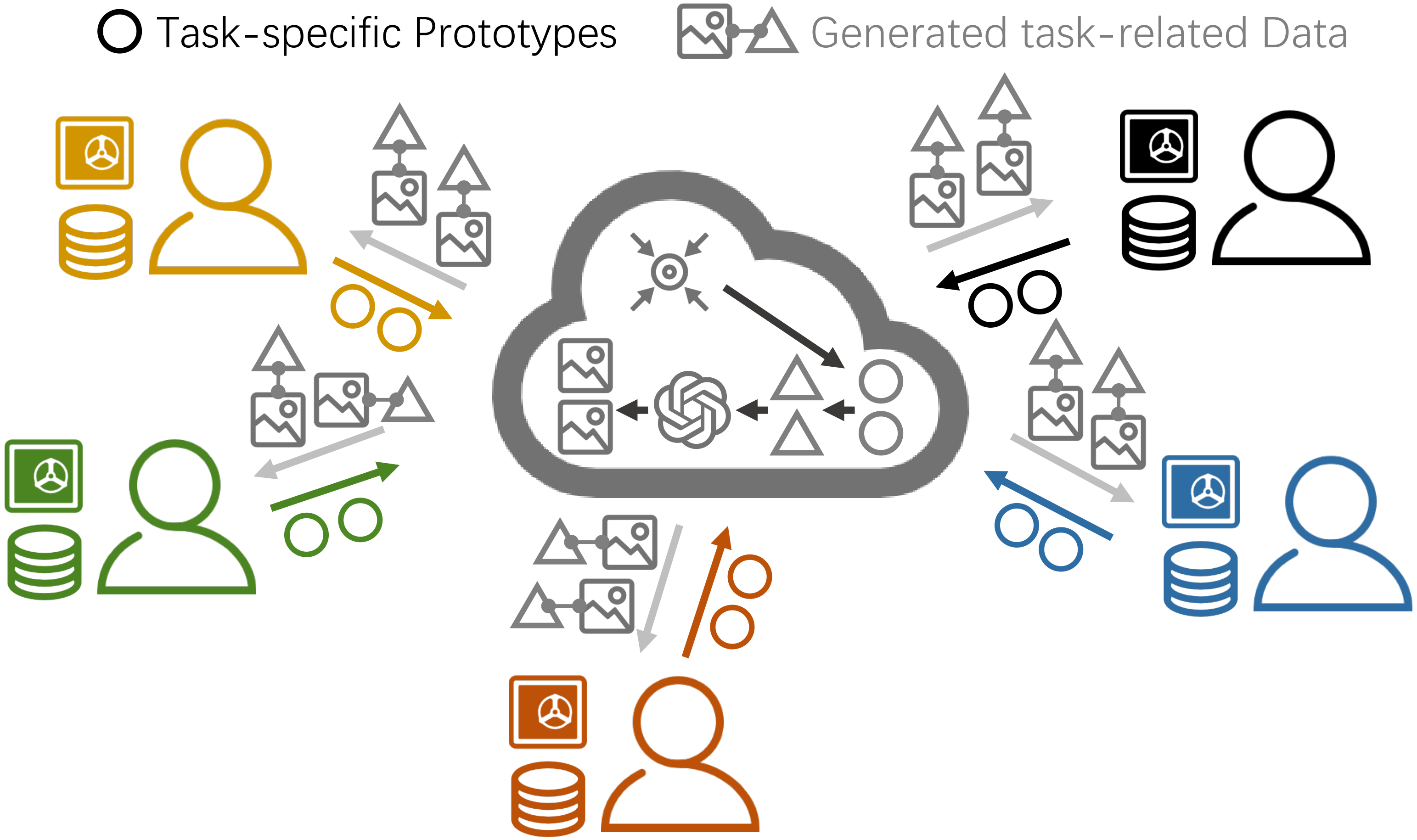


② Publications

- **[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.
  - The first one that integrates a large generative model (Stable Diffusion, etc.) into heterogeneous small models (ResNets, ViTs, etc.) collaboration

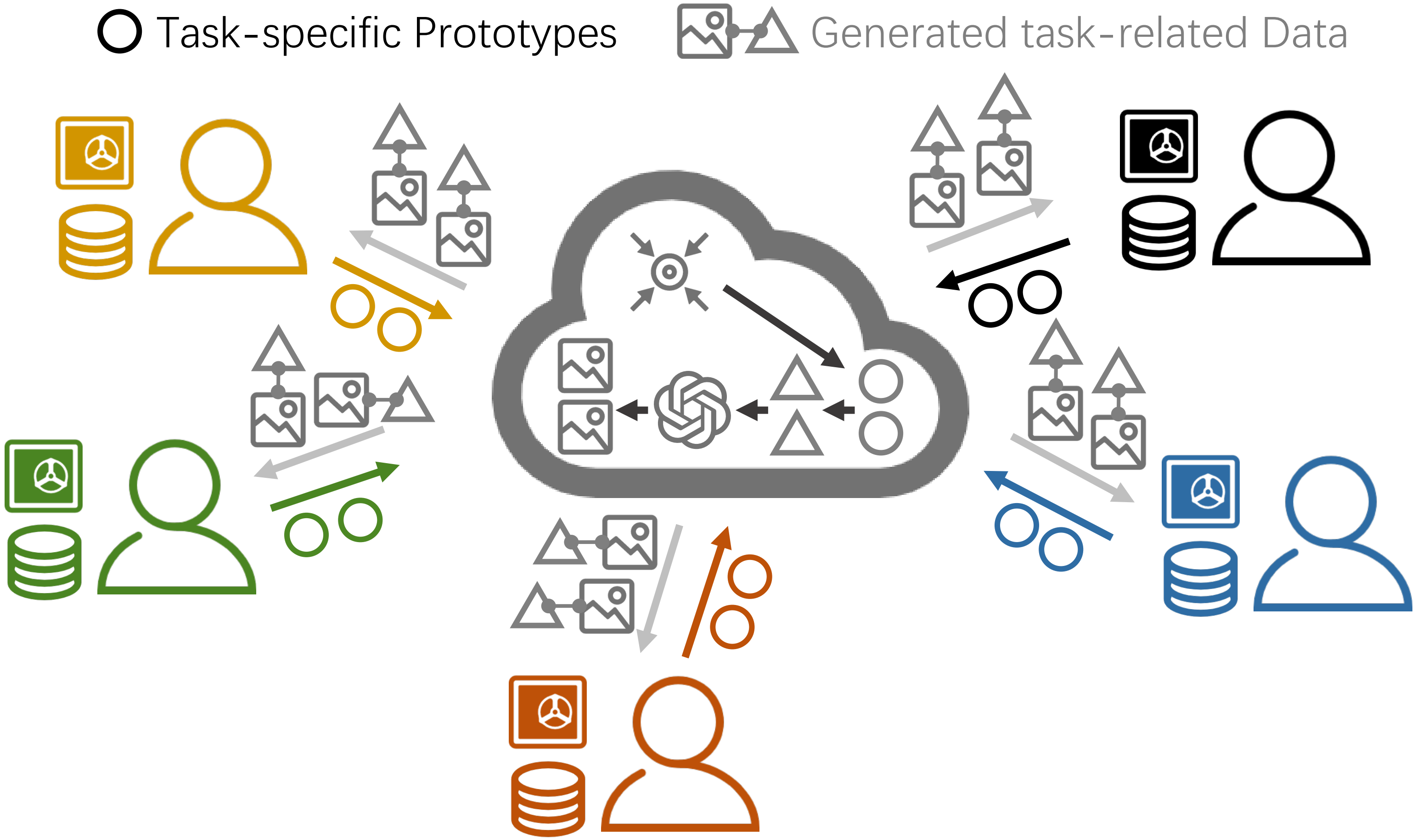
FedKTL

- 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**



Accepted by CVPR 2024

Outperform baselines by up to 7.31%

FedKTL

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

Task-specific Prototypes   Generated task-related Data

Accepted by CVPR 2024

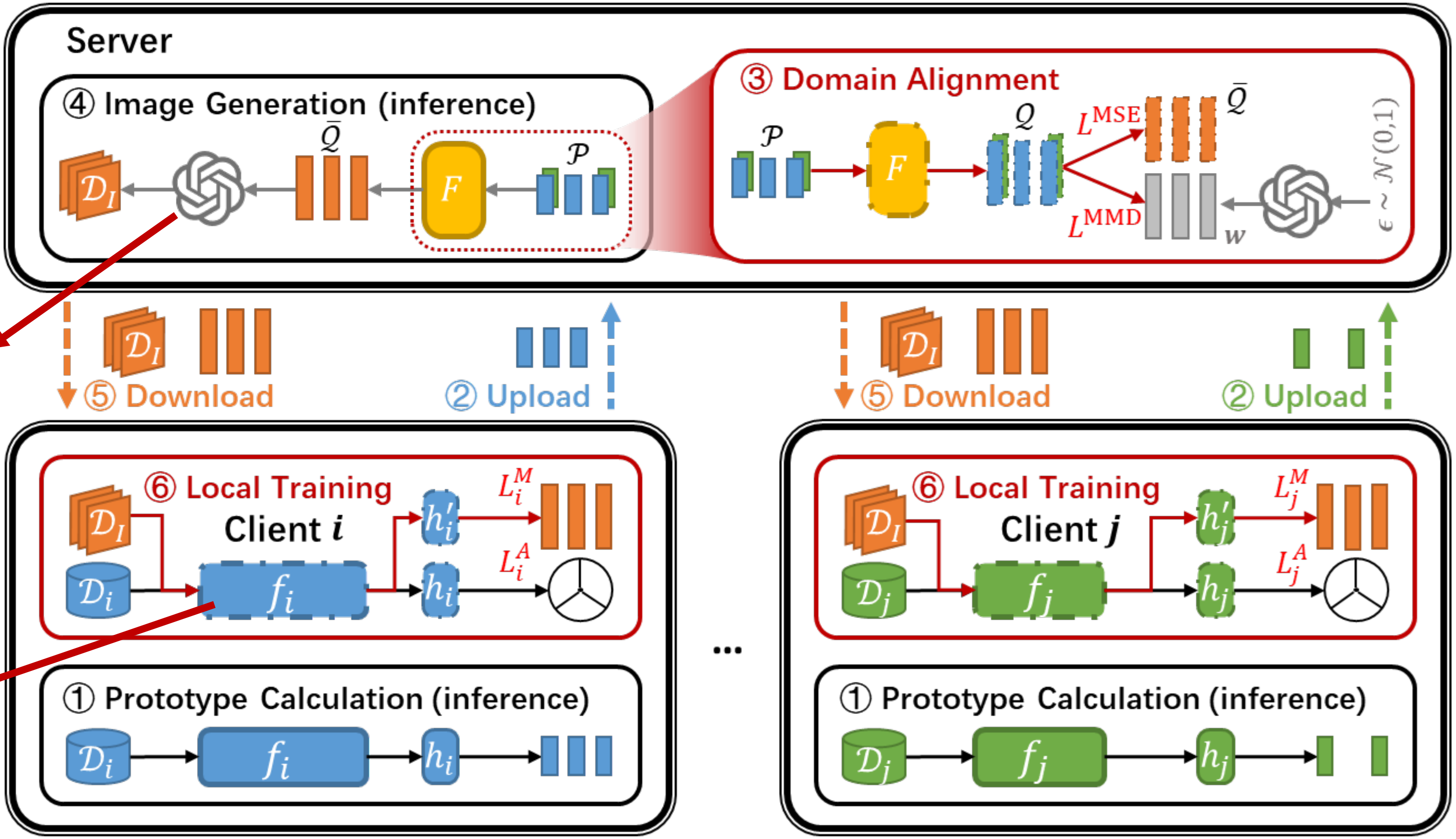
Outperform baselines by up to 7.31%

FedKTL

- **One image per class** is sufficient for FedKTL
- Transfer knowledge using an **additional supervised local task**
- **Details:**

StyleGAN series, Stable Diffusions, etc.

CNNs, ResNets, ViTs, etc.



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) Client #1 | (b) AFHQv2 (c) Benches (d) FFHQ-U (e) WikiArt

Generators pre-trained on different image datasets

	$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.5$
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

Accepted by CVPR 2024

Outperform baselines by up to 7.31%

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.

② HtFLlib on Device

- **Real-world evaluation of HtFL methods**
  - + CoLExT

- 28 Single Board Computers (SBC)
  - Orange Pi, LattePanda, Nvidia Jetson
- 20 Smartphones
  - Samsung, Xiaomi, Google Pixel, Asus ROG, One Plus
- High Voltage Power Meter
- Wired and wireless networking
- Workstation - FL Server

https://github.com/TsingZ0/HtFL-OnDevice

② HtFLlib on Device

- **Real-world evaluation of HtFL methods**
  - + CoLExT

CoLExT Experiment Deployment

Deploying experiments on CoLExT requires the specification of a CoLExT config file. An example is shown below. Additional examples will be placed inside `colext_experiments/`.

Important notes:

1. Before running the client Python code, client devices assign their own data based on the client ID provided by CoLExT, using the `./config_device_data.sh` script.
2. It's possible to specify additional arguments to a particular client group, allowing different arguments to be assigned to different groups.

Example config:

```yaml
# colext_experiments/example_config.yaml
project: htfl-ondevice

code:
  # Path to the code root dir, relative to the config file
  # Defaults to the config file dir if omitted
```

https://github.com/TsingZ0/HtFL-OnDevice

② HtFLlib on Device

- **Real-world evaluation of HtFL methods**
  - **+ Real-world datasets**

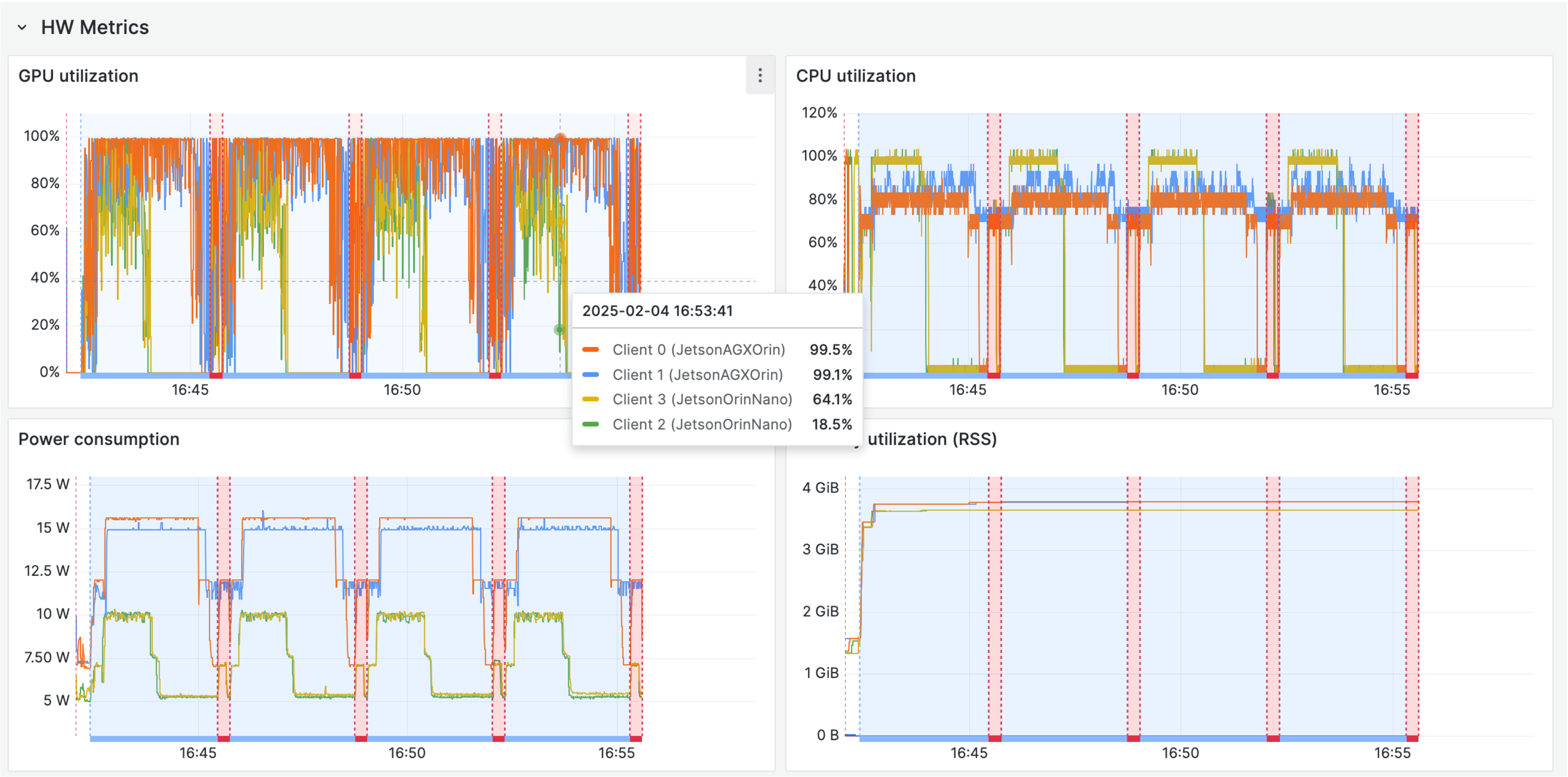
`dataset` : Realistically and naturally distributed datasets sourced generated by the codes from PFLlib. You can find more raw data here

- HAR (Human Activity Recognition) (30 clients, 6 labels)
- PAMAP2 (9 clients, 12 labels)
- iWildCam (194 camera traps, 158 labels)

https://github.com/TsingZ0/HtFL-OnDevice

② HtFLlib on Device

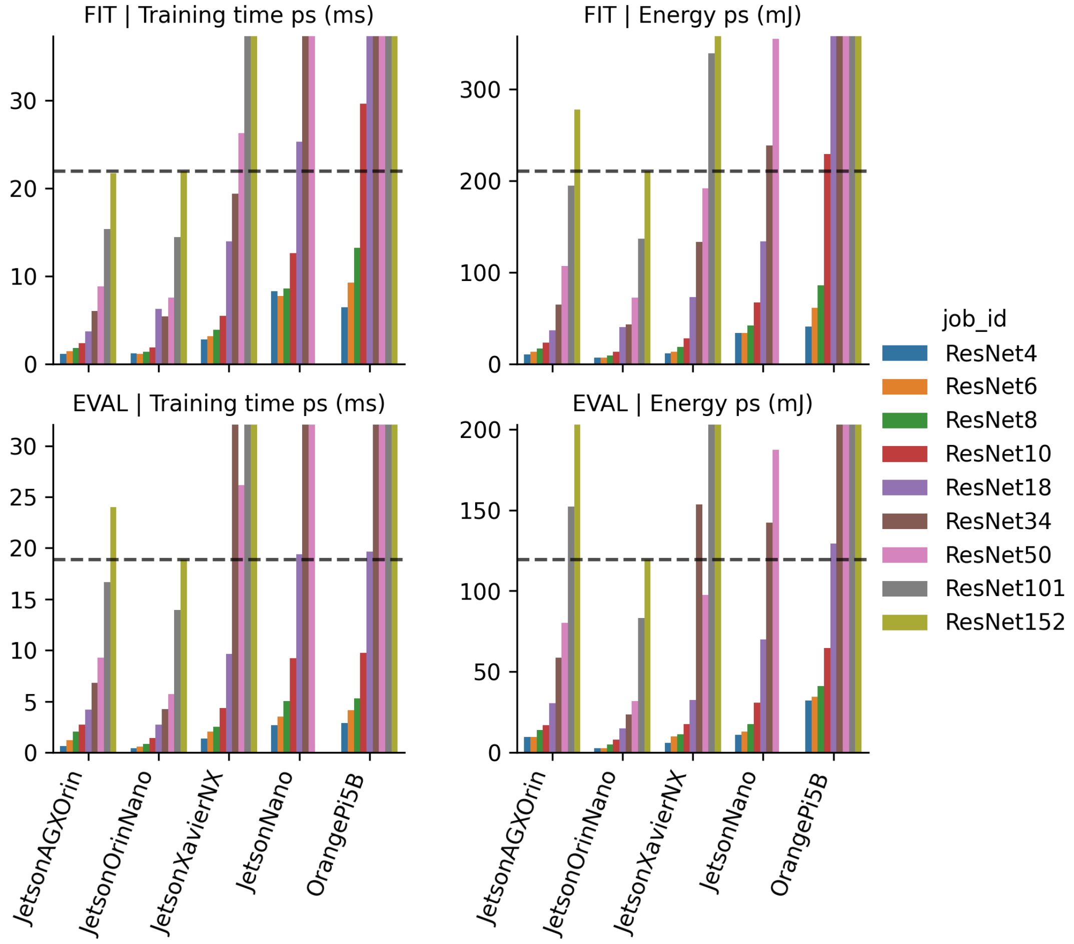
- **Real-world evaluation of HtFL methods**
  - **+ Metrics such as delay, energy consumption, and CPU/GPU utilization**

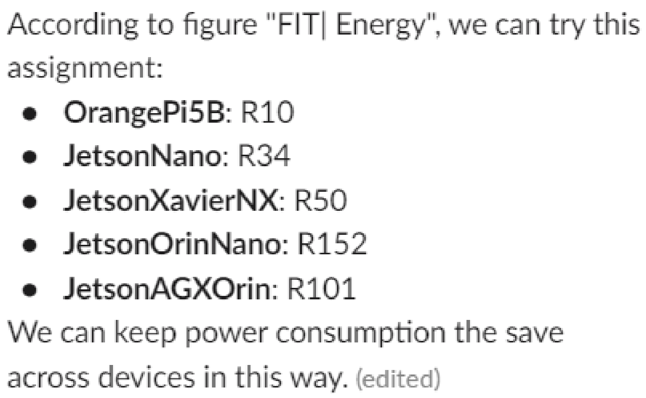


https://github.com/TsingZ0/HtFL-OnDevice

② HtFLlib on Device

- **Real-world evaluation of HtFL methods**
  - **+ Heterogeneous models assignment scheme**

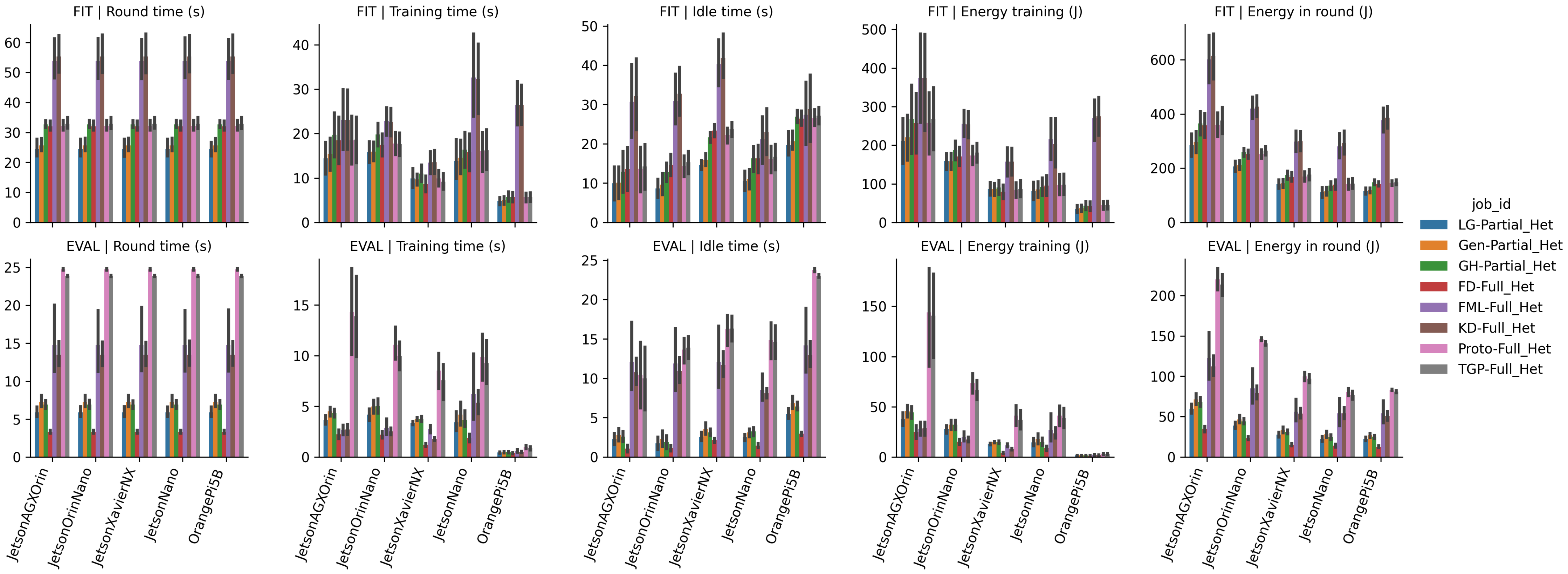




https://github.com/TsingZ0/HtFL-OnDevice

② HtFLlib on Device

- **Real-world evaluation of HtFL methods**
  - **+ Benchmarking time/energy**



https://github.com/TsingZ0/HtFL-OnDevice

Feel free to contact me!

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

Thanks!