

# An Upload-Efficient Scheme for Transferring Knowledge From a Server-Side Pre-trained Generator to Clients in Heterogeneous Federated Learning

Jianqing Zhang<sup>1</sup>

Yang Liu<sup>2</sup>

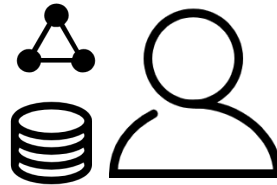
Yang Hua<sup>3</sup>

Jian Cao<sup>1</sup>



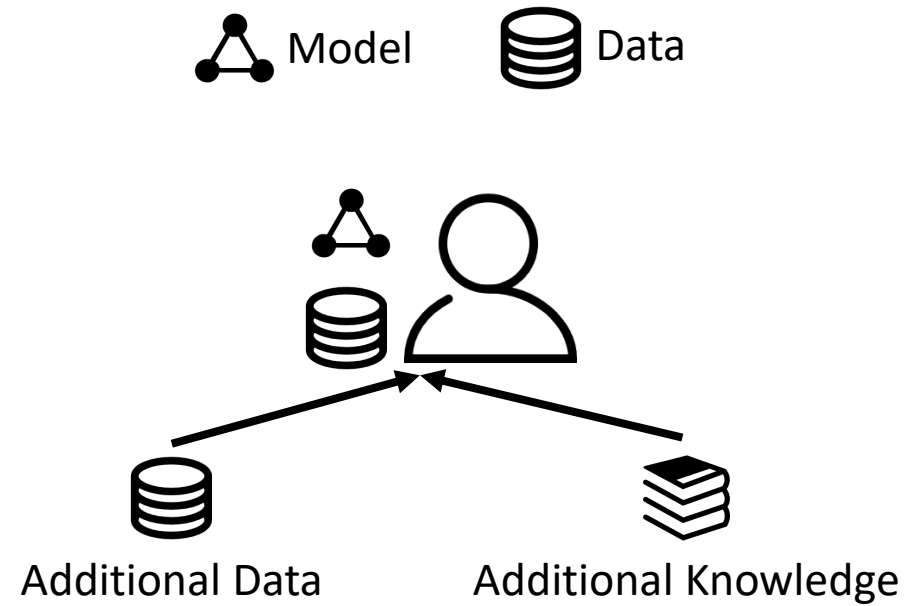
# Data shortage

- Data shortage challenges AI model training for individuals and companies.



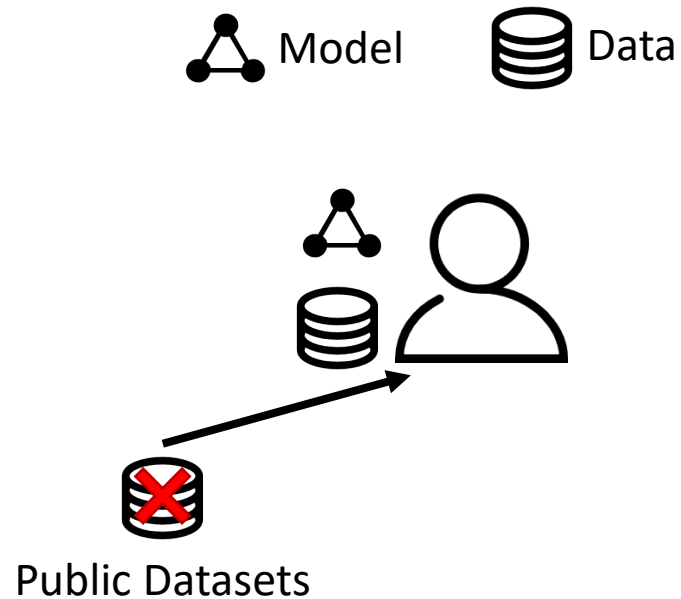
# Data shortage

- Additional **data** and **knowledge** can mitigate this challenge.



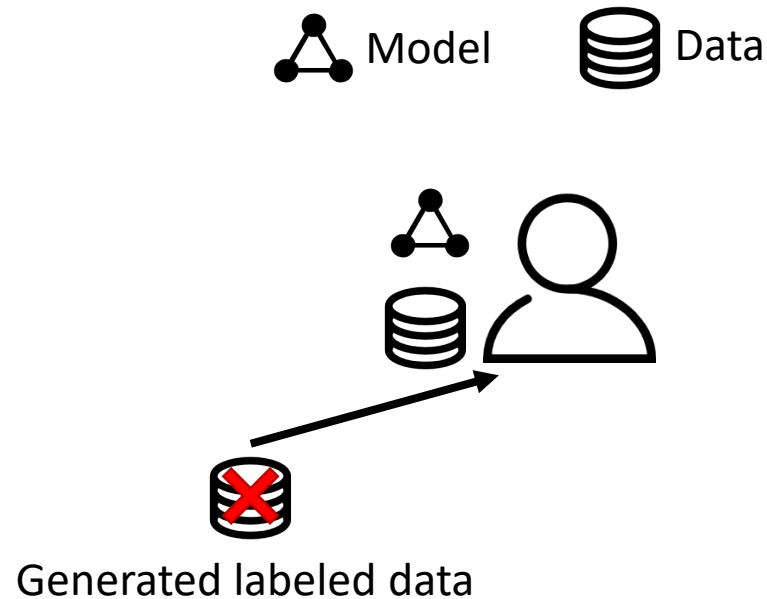
# Public datasets

- Additional data need to be **task-related**.
- It is hard to extract such data from **public datasets**.



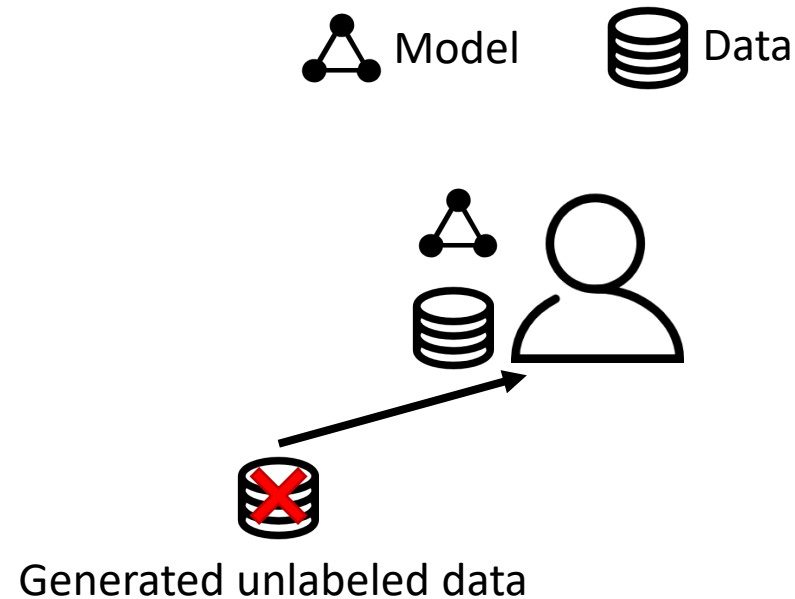
# Generated labeled data

- Transmitting human-readable information, e.g., semantics of labels, about specific tasks to the generator raises **privacy concerns**.



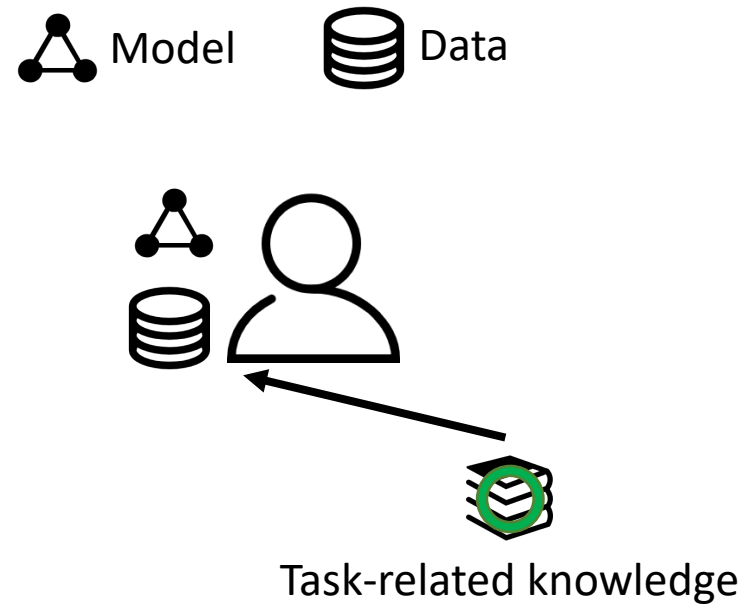
# Generated unlabeled data

- Without exposing such information, the generated **unlabeled data** belongs to the **generator's output domain**, which is not naturally related to specific tasks.
- Fulfilling unlabeled data is **challenging** in deep learning.



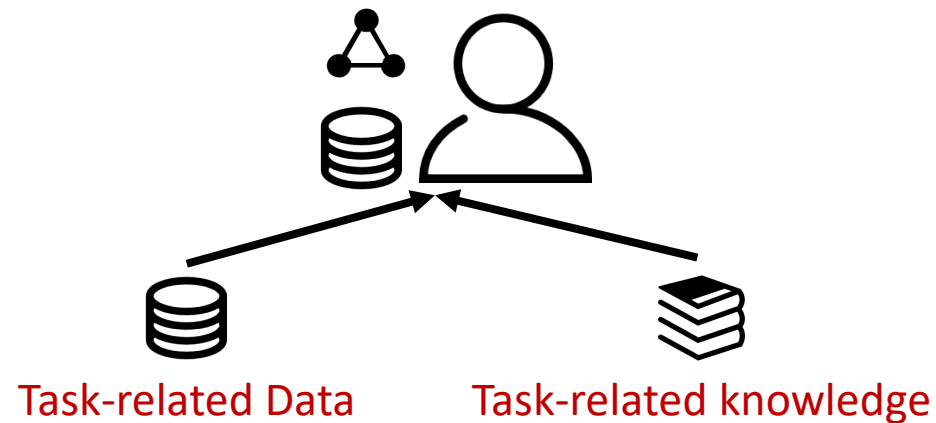
# Knowledge from others

- Additional knowledge need to be **task-related**.
- Clients in federated learning (FL) intend to solve **similar tasks**, so we use FL techniques.



# Our method

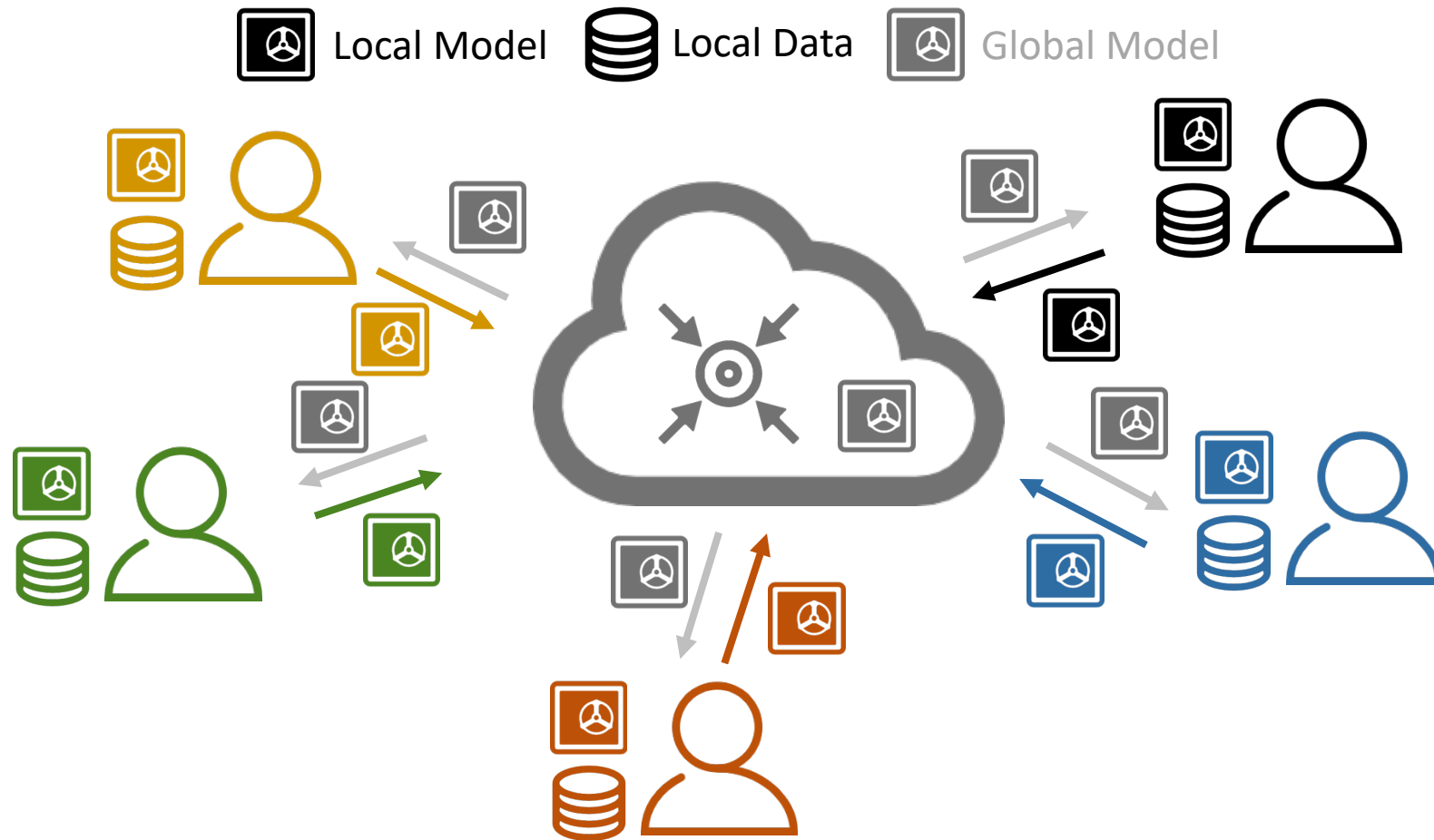
- Propose a **federated learning (FL) method** to share **task-related (abstract) knowledge**.
- **Adapt a pre-trained generator** to produce **task-related data** based on task-related knowledge.
- **Transfer** task-related knowledge and data **to each client** via **an additional supervised task**.





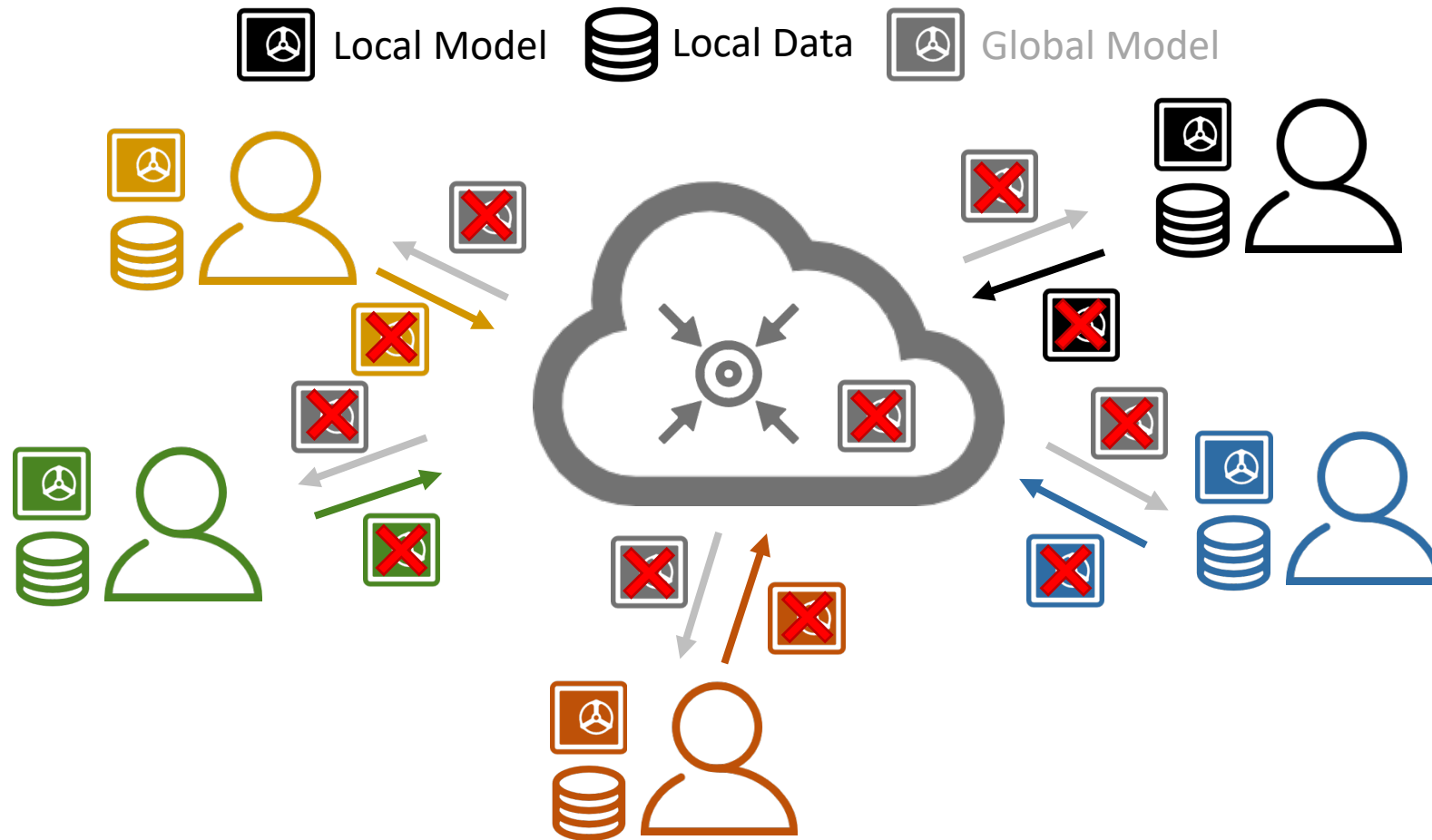
# Heterogeneous Federated Learning (HtFL)

- Data heterogeneity, model heterogeneity, communication cost, **intellectual property**, etc.



# Heterogeneous Federated Learning (HtFL)

- The **intellectual property** is overlooked by most previous work.
- To protect intellectual property, we **cannot expose model parameters** among clients.



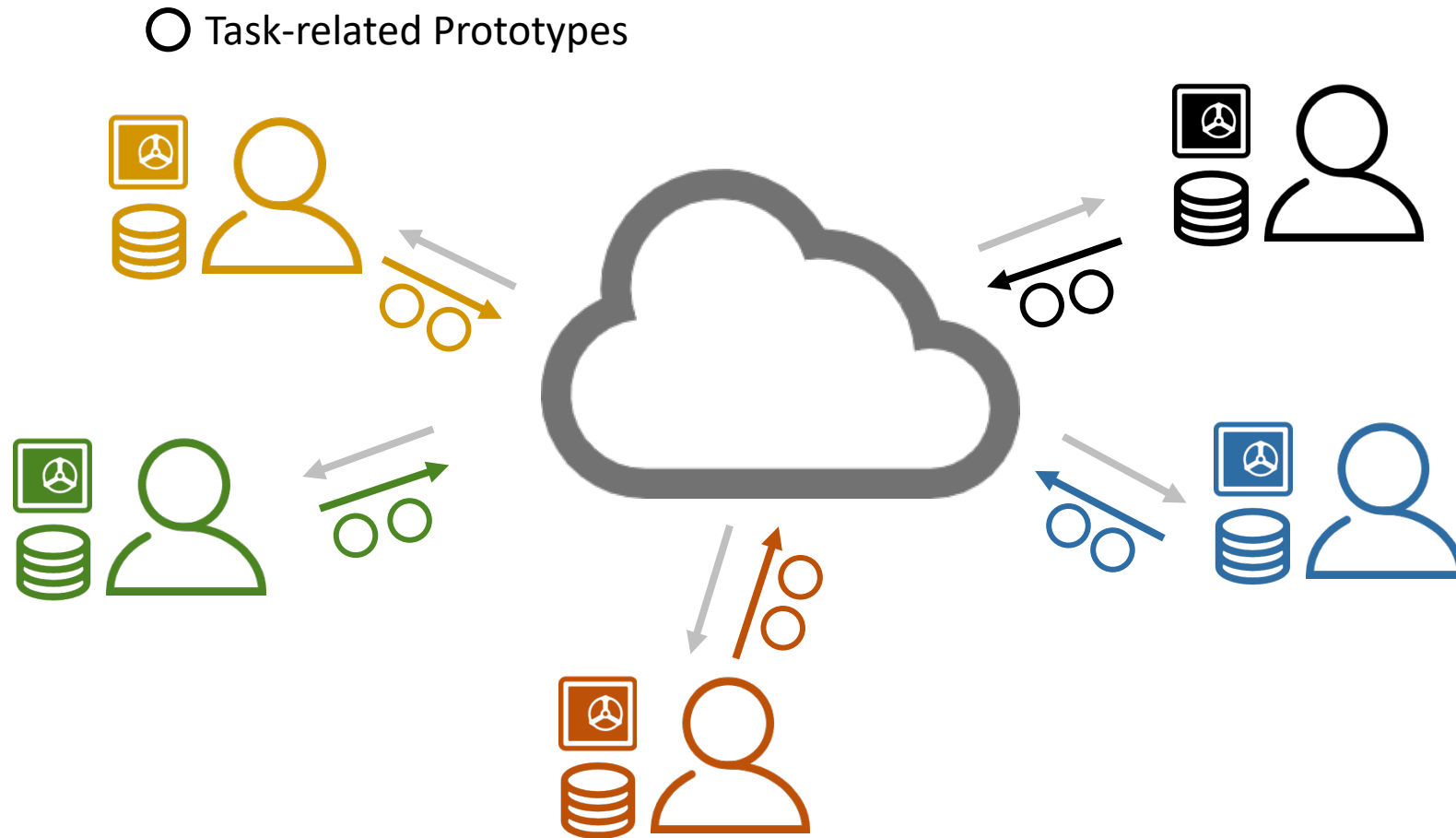
# Heterogeneous Federated Learning (HtFL)

- Transmit **lightweight knowledge carriers** instead of exposing model parameters among clients



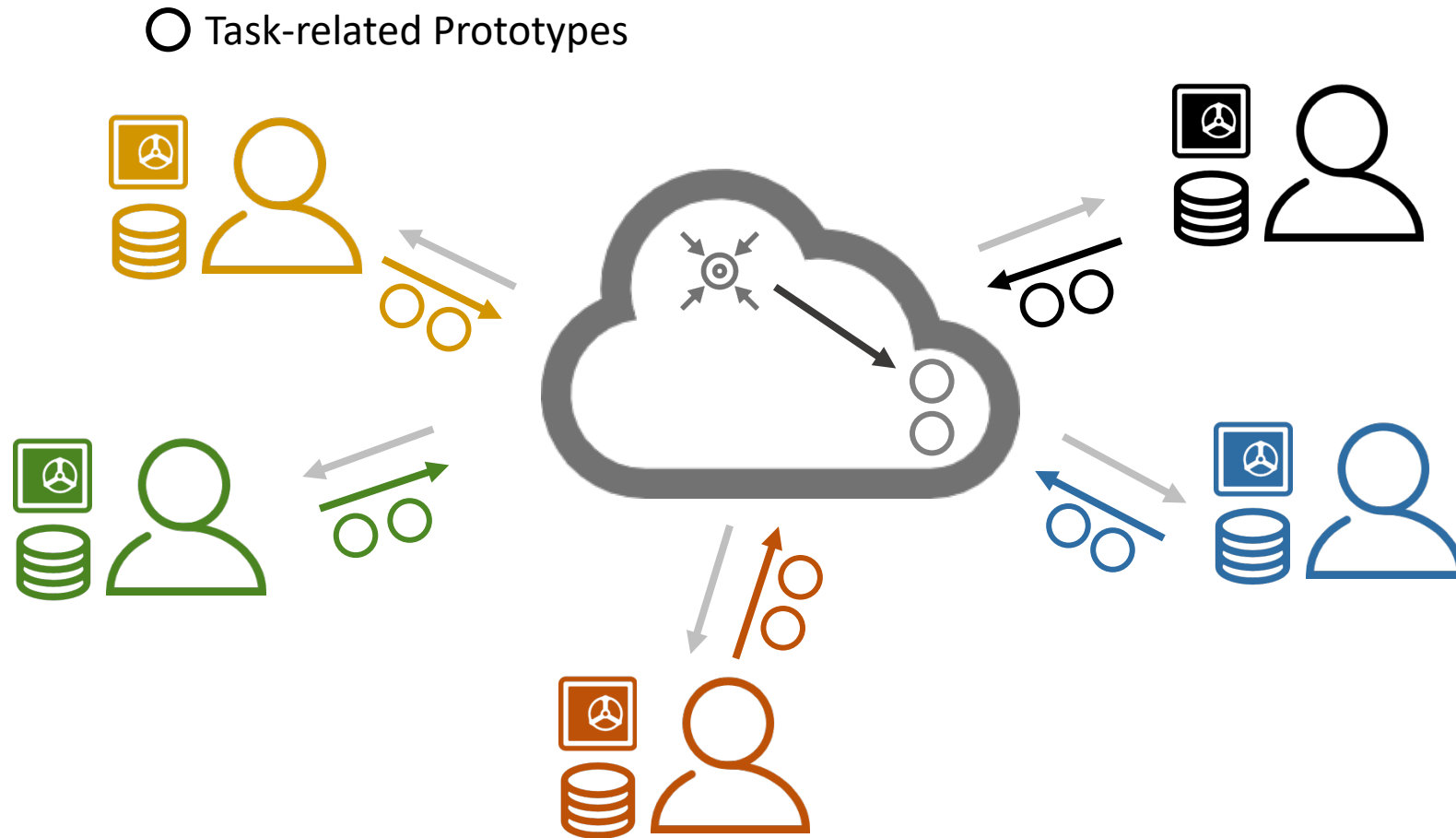
# Task-related prototypes

- Specifically, in our work, clients upload **task-related** prototypes  $\bigcirc$  to the server.



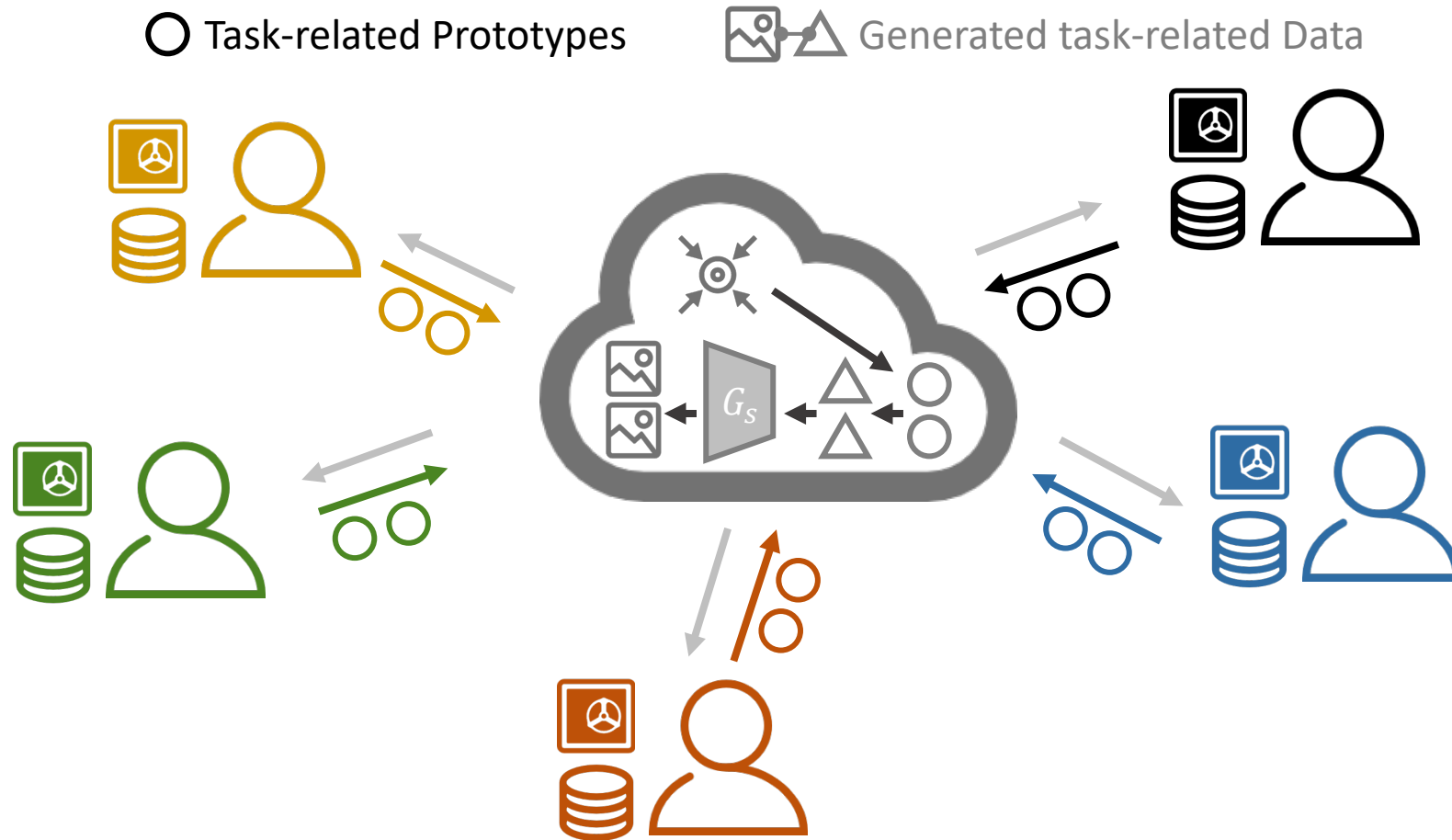
# Prototype aggregation

- The server then aggregates client prototypes.





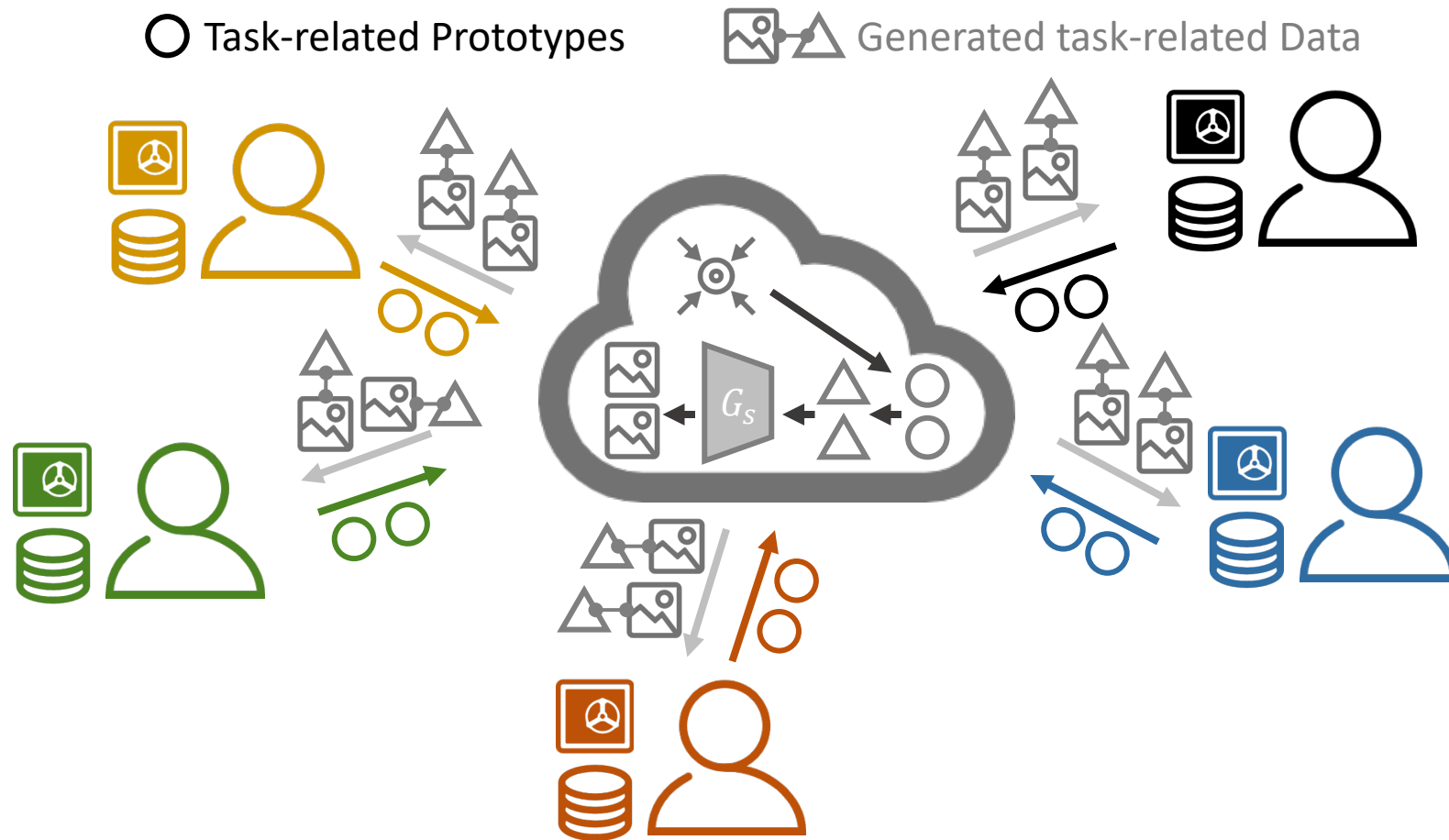
# Image generation

- The server maps  $\bigcirc$  to **latent vectors**  $\triangle$ , and generates images .

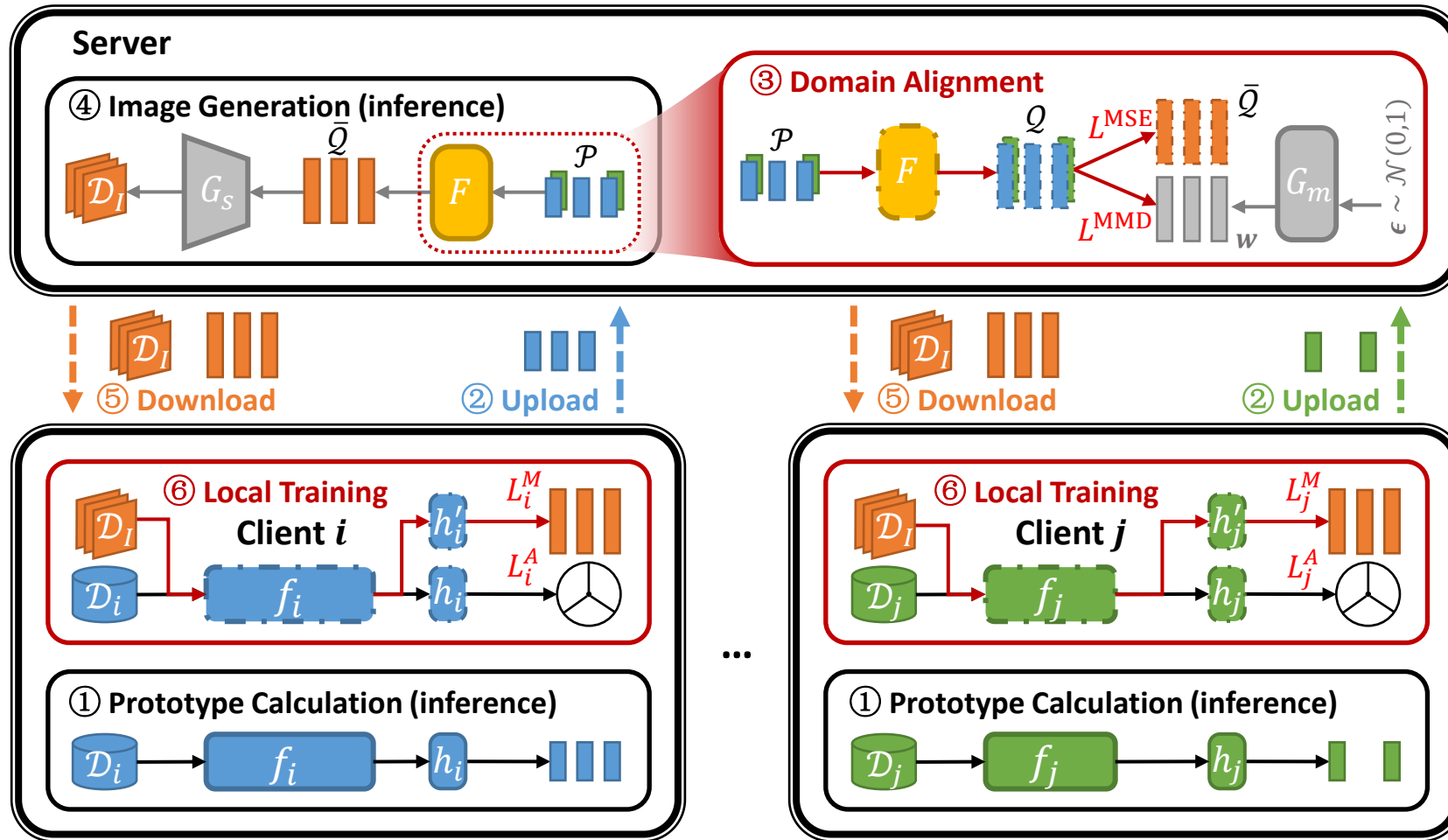


# Image-vector pairs

- The server sends **image-vector pairs**   to each client for an **additional supervised task**.

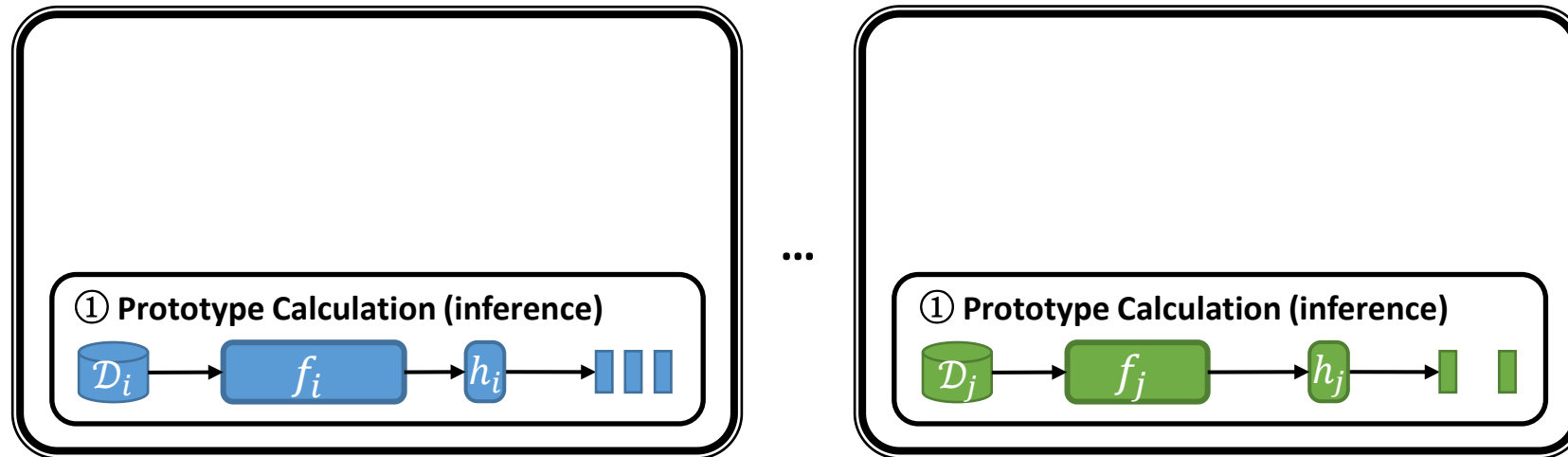
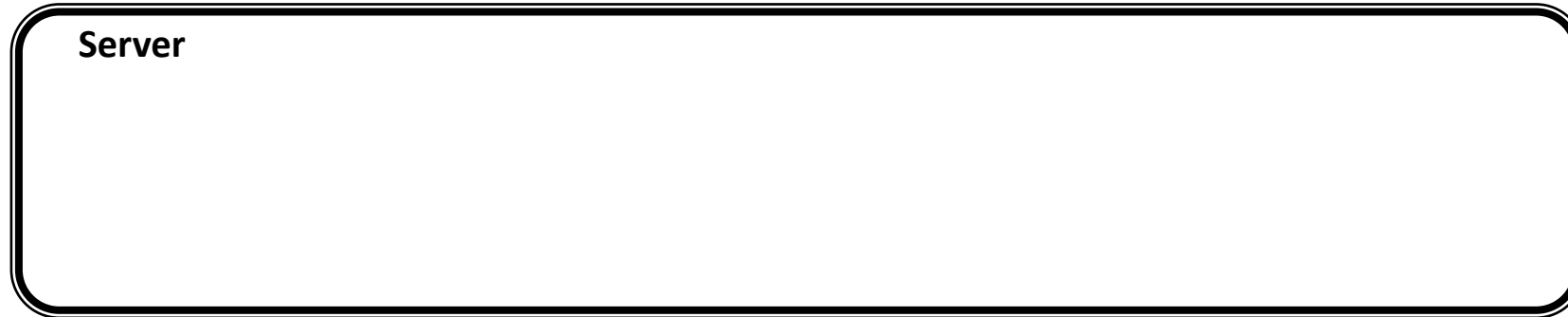


# Federated Knowledge-Transfer-Loop (FedKTL)

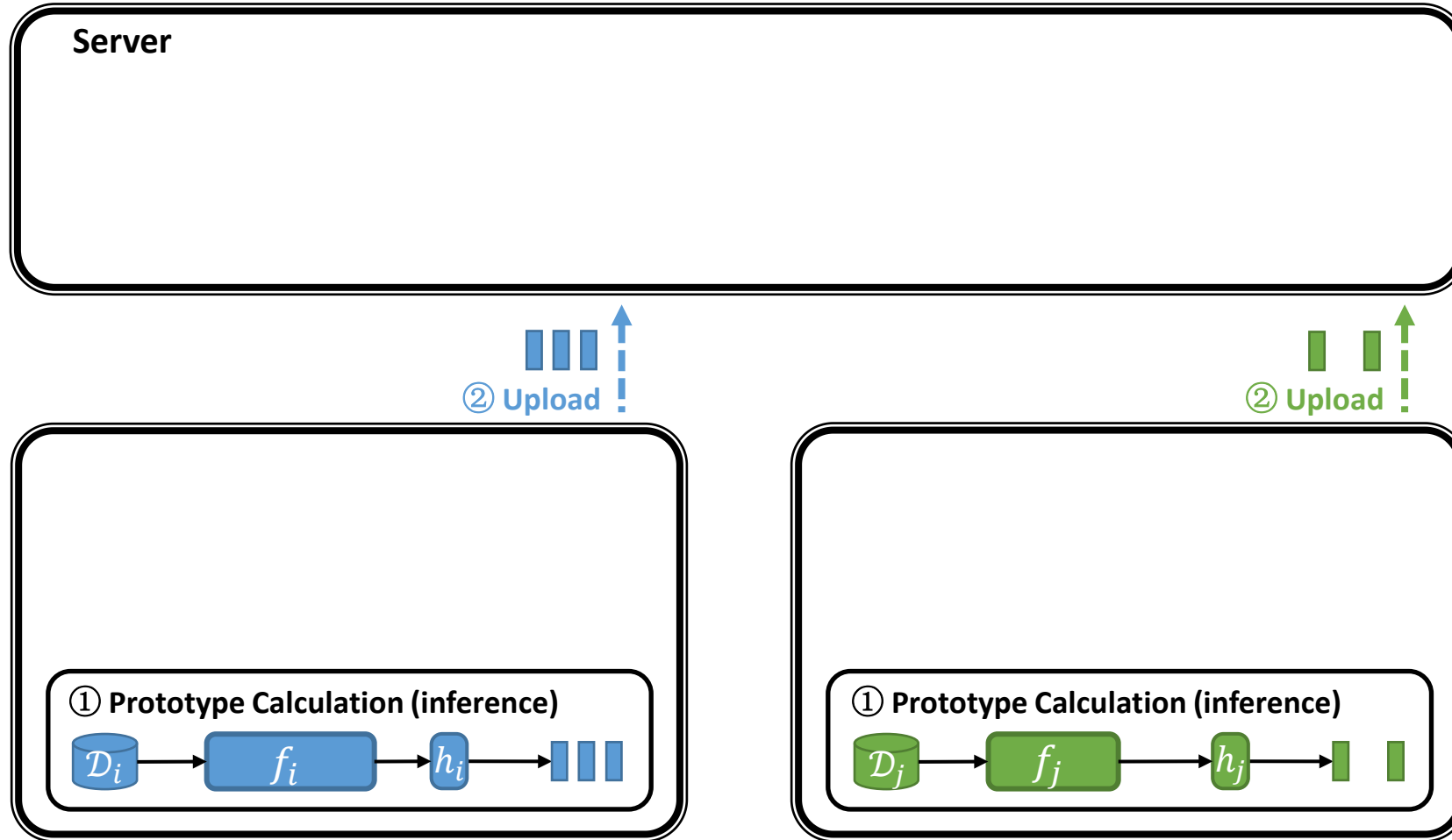




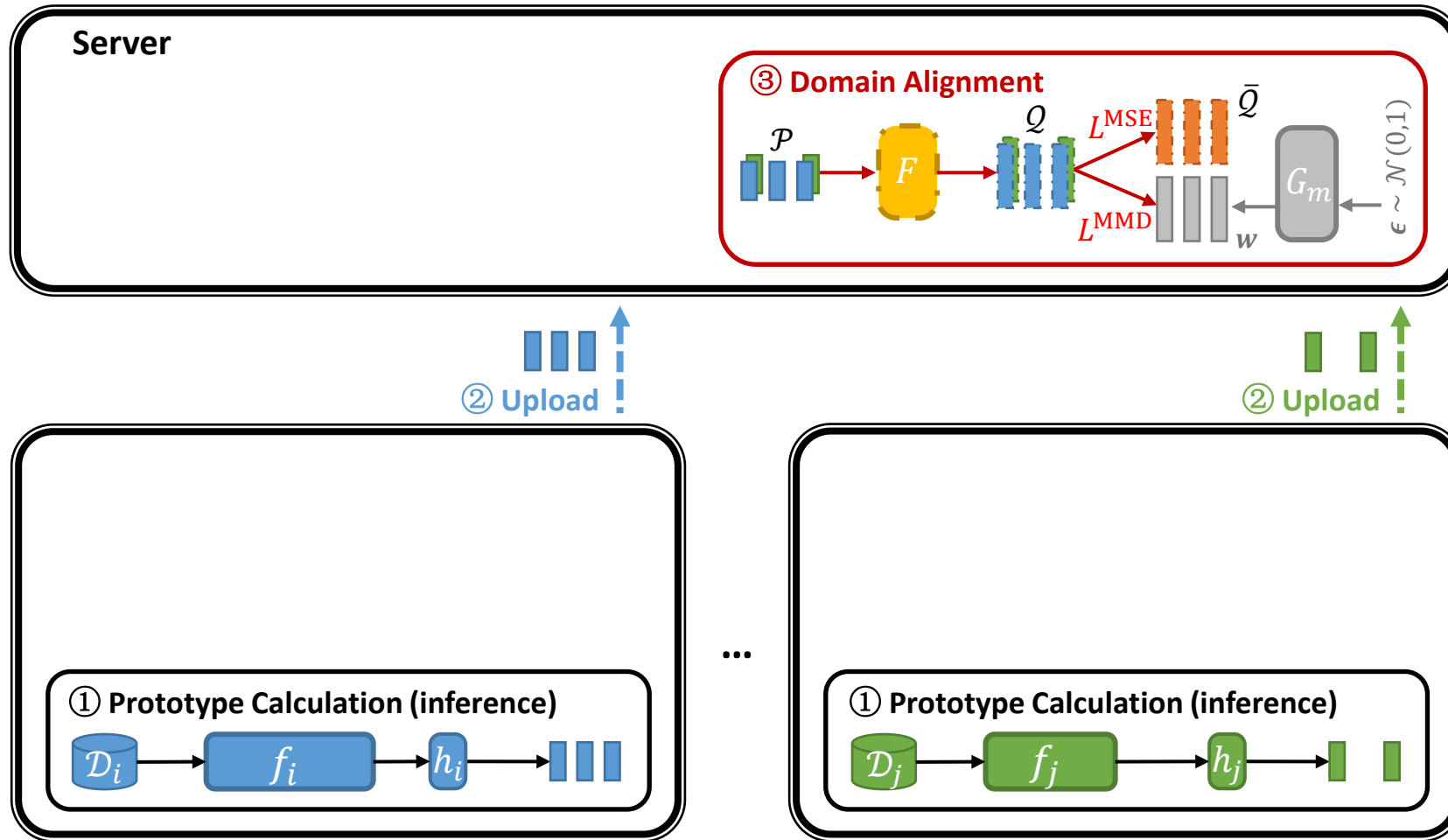
# ① Prototype Calculation (inference)



## ② Upload

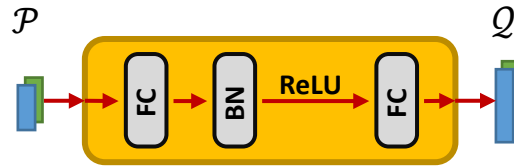


### ③ Domain Alignment



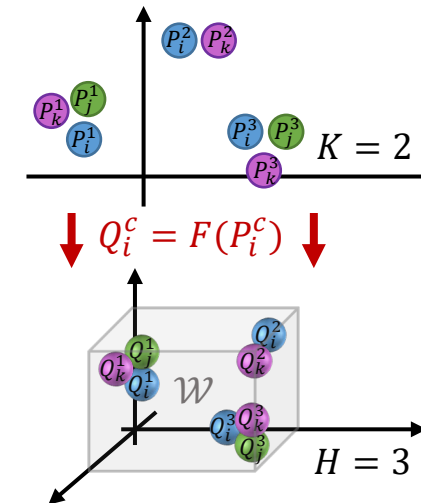
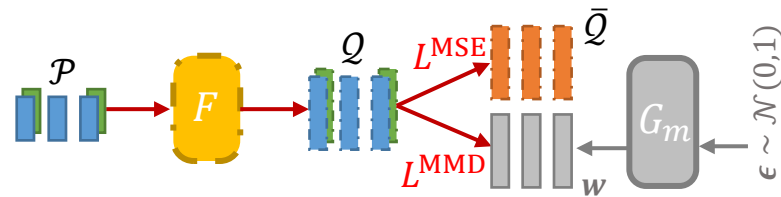
### ③ Domain Alignment

- The architecture of the feature transformer  $F$ .

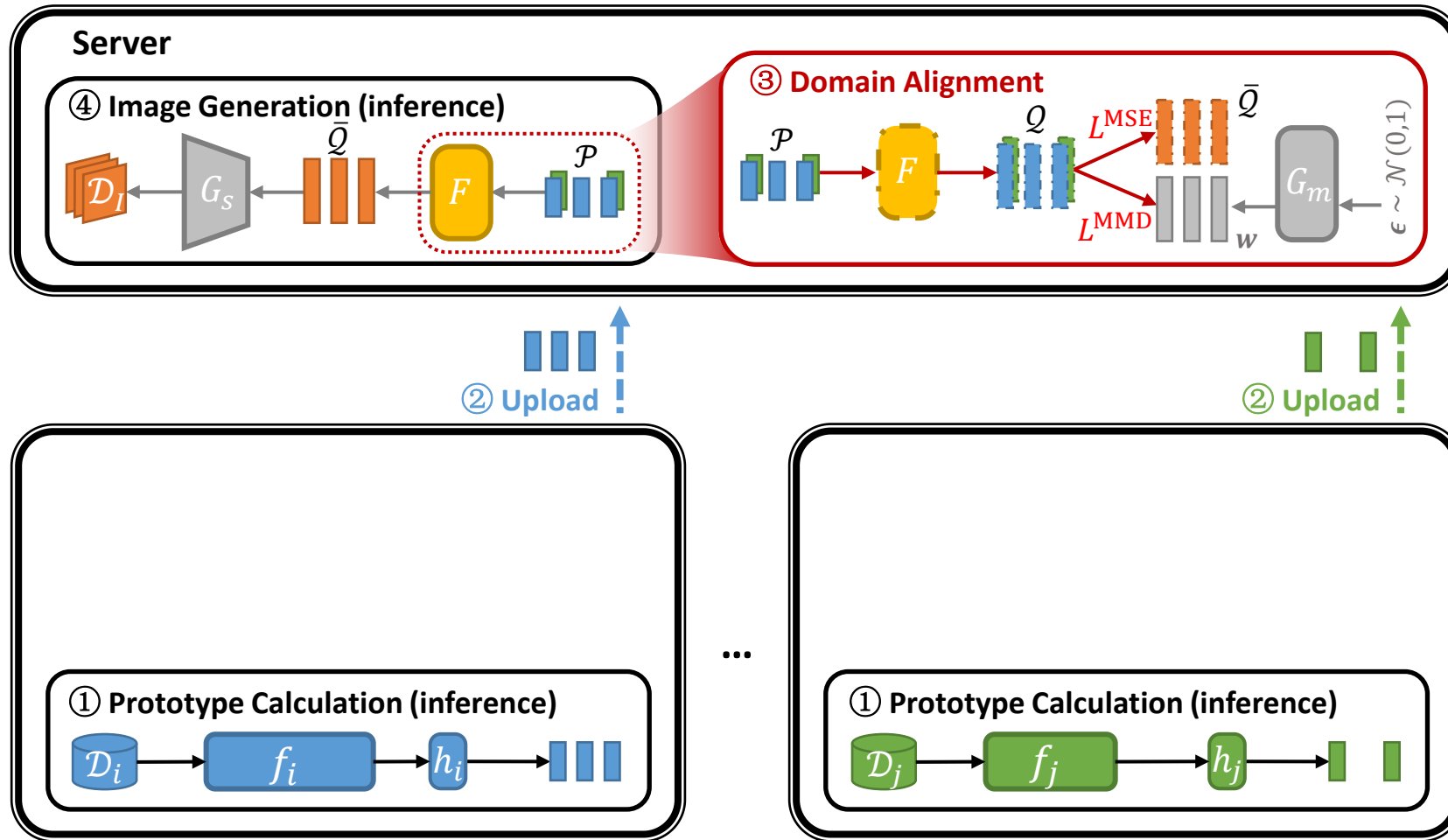


### ③ Domain Alignment

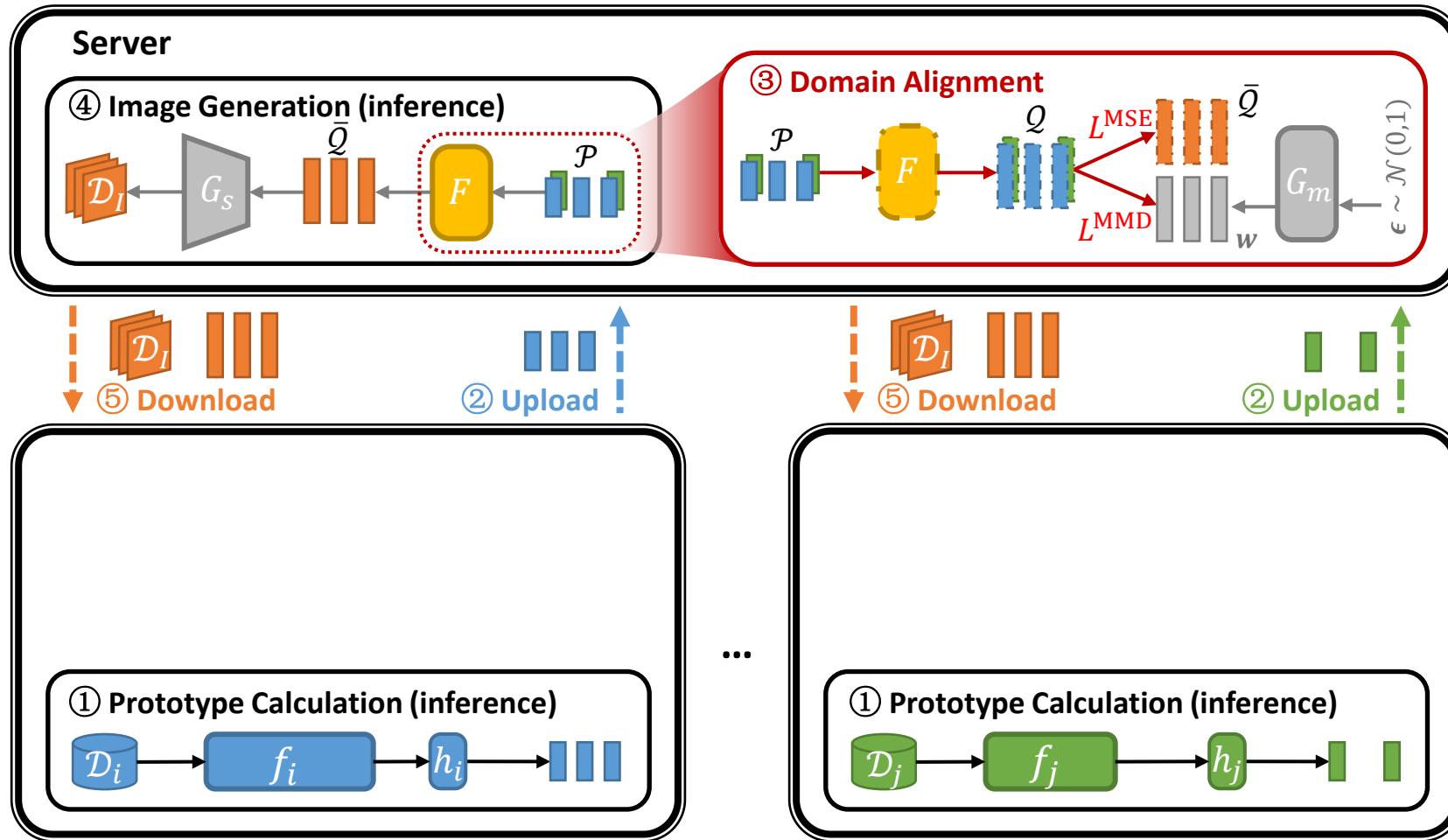
- A domain alignment example.



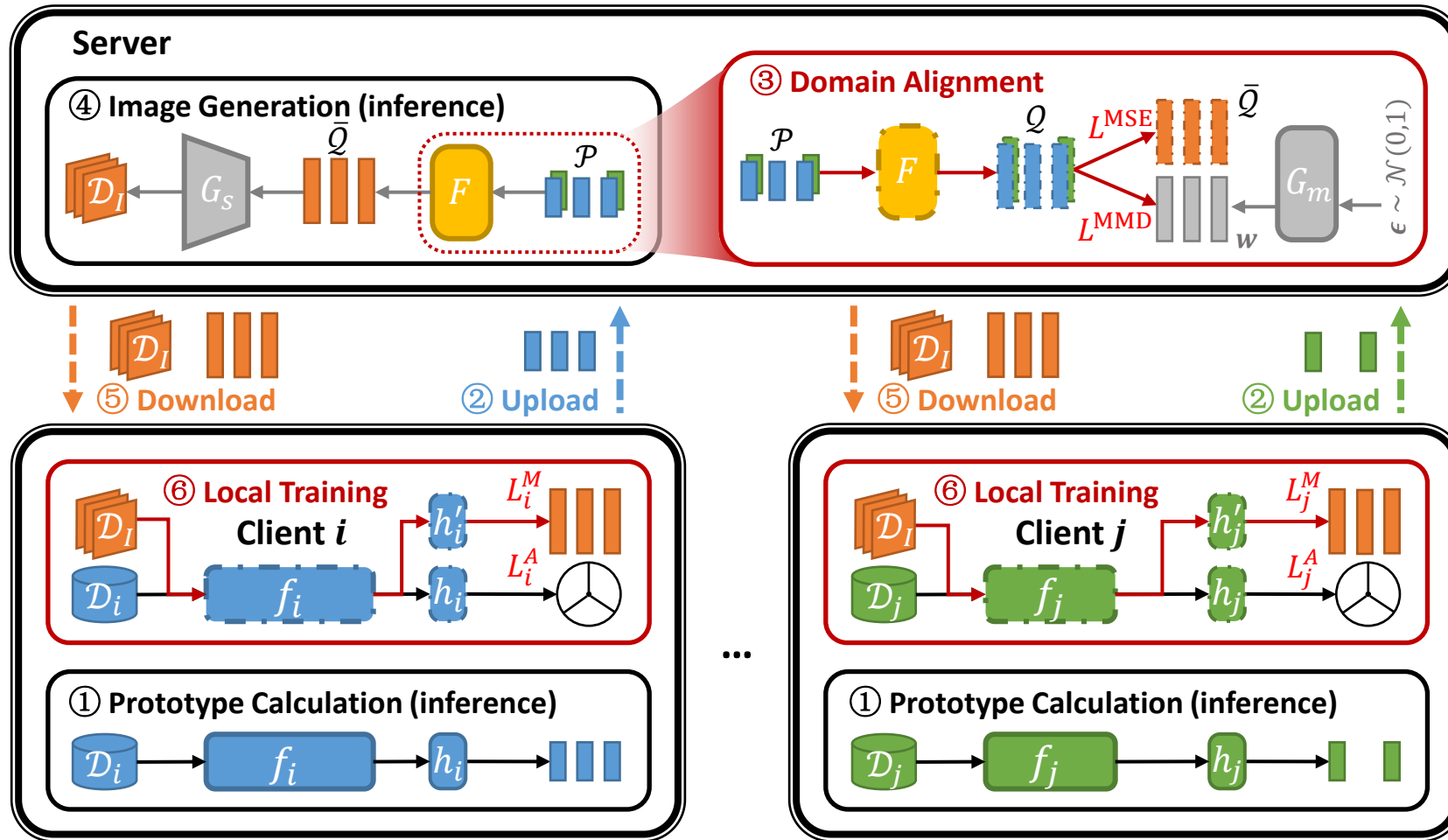
## ④ Image Generation (inference)



# ⑤ Download



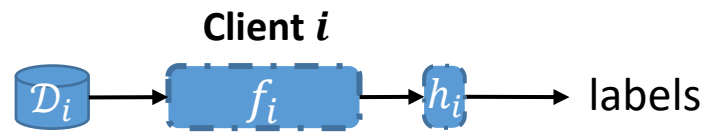
# ⑥ Local Training





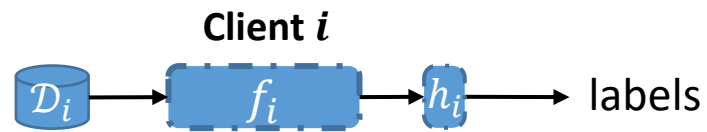
## ⑥ Local Training

- Original local task: **classification**.



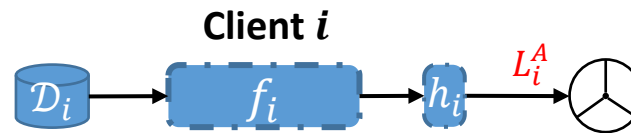
## ⑥ Local Training

- Heterogeneous models produce **biased prototypes** due to their divergent capabilities.



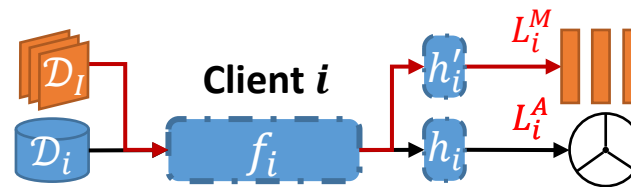
## ⑥ Local Training

- Replace the original classifier part by an **ETF classifier**[1] to produce unbiased prototypes.



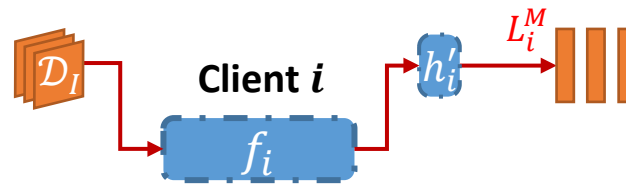
## ⑥ Local Training

- Transfer task-related knowledge and data to clients through an **additional supervised task**.



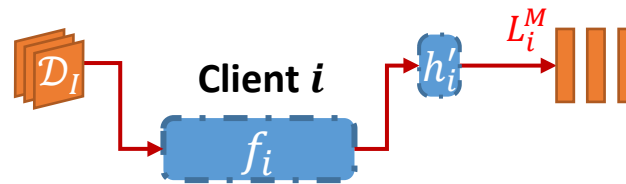
## ⑥ Local Training

- The image-vector pairs brings both **common (from the pre-trained generator)** and **shared (from participating clients)** knowledge to the **feature extractor part** of each client model.



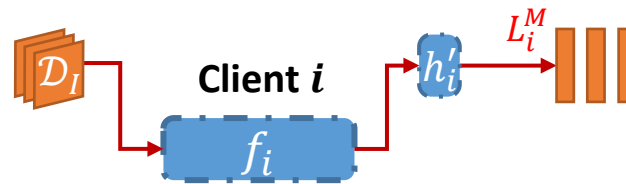
## ⑥ Local Training

- We only transfer knowledge to **enhance the general feature extraction capability**.



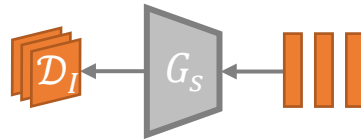
## ⑥ Local Training

- Thus, the **semantic relationship** between the generated images and local data is **insignificant**.



# Support for various pre-trained generators

- In other words, **generators pre-trained on any image datasets are applicable.**





# Support for various pre-trained generators

- In other words, **generators pre-trained on any image datasets are applicable.**



(a) Client #1



(b) AFHQv2



(c) BENCHES



(d) FFHQ-U



(e) WikiArt

# Support for various pre-trained generators

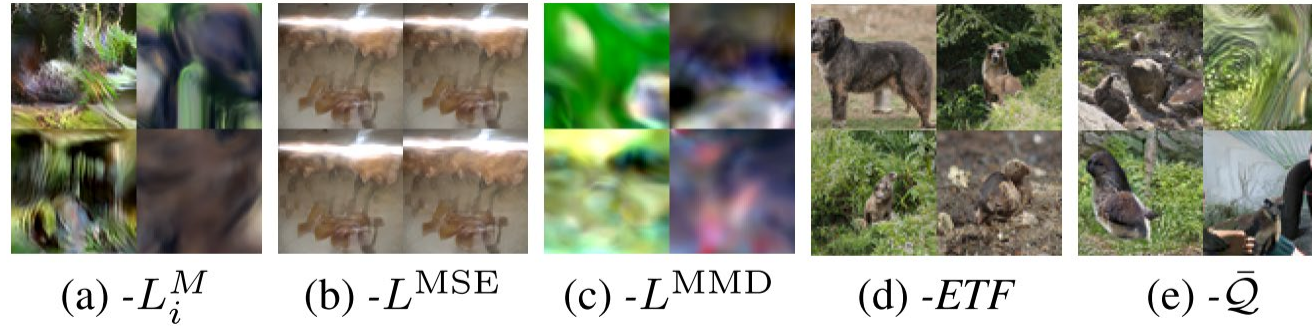
- In other words, **generators pre-trained on any image datasets are applicable.**

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

Table 6. The test accuracy (%) on Tiny-ImageNet in the practical setting using HtFE<sub>8</sub> with different pre-trained StyleGAN3s, which are represented by the names of the pre-training datasets.

# Ablation study

- Each component plays a vital role, and none can be omitted.



# Excellent performance

- Experiment on four datasets.

Settings	Pathological Setting				Practical Setting			
Datasets	Cifar10	Cifar100	Flowers102	Tiny-ImageNet	Cifar10	Cifar100	Flowers102	Tiny-ImageNet
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	<b>88.43±0.13</b>	<b>62.01±0.28</b>	<b>64.72±0.62</b>	<b>34.74±0.17</b>	<b>87.63±0.07</b>	<b>46.94±0.23</b>	<b>53.16±0.08</b>	<b>28.17±0.18</b>

Table 1. The test accuracy (%) on four datasets in the pathological and practical settings using HtFE<sub>8</sub>.

# Excellent performance

- Experiment on four datasets using 14 kinds of models including **CNNs and ViTs**.

Settings	Different Degrees of Model Heterogeneity					Large Client Amount ( $\rho = 0.5$ )		
	HtFE <sub>2</sub>	HtFE <sub>3</sub>	HtFE <sub>4</sub>	HtFE <sub>9</sub>	HtM <sub>10</sub>	50 Clients	100 Clients	200 Clients
LG-FedAvg	46.61±0.24	45.56±0.37	43.91±0.16	42.04±0.26	—	37.81±0.12	35.14±0.47	27.93±0.04
FedGen	43.92±0.11	43.65±0.43	40.47±1.09	40.28±0.54	—	37.95±0.25	34.52±0.31	28.01±0.24
FedGH	46.70±0.35	45.24±0.23	43.29±0.17	43.02±0.86	—	37.30±0.44	34.32±0.16	29.27±0.39
FML	45.94±0.16	43.05±0.06	43.00±0.08	42.41±0.28	39.87±0.09	38.47±0.14	36.09±0.28	30.55±0.52
FedKD	46.33±0.24	43.16±0.49	43.21±0.37	42.15±0.36	40.36±0.12	38.25±0.41	35.62±0.55	31.82±0.50
FedDistill	46.88±0.13	43.53±0.21	43.56±0.14	42.09±0.20	40.95±0.04	38.51±0.36	36.06±0.24	31.26±0.13
FedProto	43.97±0.18	38.14±0.64	34.67±0.55	32.74±0.82	36.06±0.10	33.03±0.42	28.95±0.51	24.28±0.46
<b>FedKTL</b>	<b>48.06±0.19</b>	<b>49.83±0.44</b>	<b>47.06±0.21</b>	<b>50.33±0.35</b>	<b>45.84±0.15</b>	<b>43.16±0.82</b>	<b>39.73±0.87</b>	<b>34.24±0.45</b>

Table 2. The test accuracy (%) on Cifar100 in the practical setting with different degrees of model heterogeneity or large client amounts.

# Excellent performance

- Our FedKTL outperforms counterparts by up to **7.31%**.

Settings	Different Degrees of Model Heterogeneity					Large Client Amount ( $\rho = 0.5$ )		
	HtFE <sub>2</sub>	HtFE <sub>3</sub>	HtFE <sub>4</sub>	HtFE <sub>9</sub>	HtM <sub>10</sub>	50 Clients	100 Clients	200 Clients
LG-FedAvg	46.61±0.24	45.56±0.37	43.91±0.16	42.04±0.26	—	37.81±0.12	35.14±0.47	27.93±0.04
FedGen	43.92±0.11	43.65±0.43	40.47±1.09	40.28±0.54	—	37.95±0.25	34.52±0.31	28.01±0.24
FedGH	46.70±0.35	45.24±0.23	43.29±0.17	43.02±0.86	—	37.30±0.44	34.32±0.16	29.27±0.39
FML	45.94±0.16	43.05±0.06	43.00±0.08	42.41±0.28	39.87±0.09	38.47±0.14	36.09±0.28	30.55±0.52
FedKD	46.33±0.24	43.16±0.49	43.21±0.37	42.15±0.36	40.36±0.12	38.25±0.41	35.62±0.55	31.82±0.50
FedDistill	46.88±0.13	43.53±0.21	43.56±0.14	42.09±0.20	40.95±0.04	38.51±0.36	36.06±0.24	31.26±0.13
FedProto	43.97±0.18	38.14±0.64	34.67±0.55	32.74±0.82	36.06±0.10	33.03±0.42	28.95±0.51	24.28±0.46
FedKTL	<b>48.06±0.19</b>	<b>49.83±0.44</b>	<b>47.06±0.21</b>	<b>50.33±0.35</b>	<b>45.84±0.15</b>	<b>43.16±0.82</b>	<b>39.73±0.87</b>	<b>34.24±0.45</b>

Table 2. The test accuracy (%) on Cifar100 in the practical setting with different degrees of model heterogeneity or large client amounts.

# Excellent performance

- Our FedKTL is **upload-efficient** (lowest upload communication cost)

	Upload	Download	Accuracy
LG-FedAvg	1.03M	1.03M	$40.65 \pm 0.07$
FedGen	1.03M	7.66M	$38.73 \pm 0.14$
FedGH	0.46M	1.03M	$40.99 \pm 0.51$
FML	18.50M	18.50M	$39.86 \pm 0.25$
FedKD	16.52M	16.52M	$40.56 \pm 0.31$
FedDistill	0.09M	0.20M	$41.54 \pm 0.08$
FedProto	0.46M	1.02M	$36.34 \pm 0.28$
FedKTL	<b>0.09M</b>	7.17M	<b><math>46.94 \pm 0.23</math></b>

Table 5. The upload and download overhead per iteration using HtFE<sub>8</sub> on Cifar100 with 20 clients in the practical setting. “M” is short for million. The accuracy column is referred from Tab. 1.

# Using Stable Diffusion

- Several concepts in **various generators**, such as **StyleGAN** and **Stable Diffusion**, share similarities when generating contents, thus **they are all applicable** in our FedKTL.

Generator	StyleGAN-XL	Stable Diffusion
Accuracy	87.63	87.71

Table 8. The test accuracy (%) of our FedKTL with different pre-trained generators on Cifar10 in the practical setting using HtFE<sub>8</sub>.



# The cloud-edge scenario

- Our **knowledge transfer scheme (KTL)** is also applicable in scenarios with **only one edge client**.
  - No collaboration
  - Few-shot learning

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>

Paper with code: <https://github.com/TsingZ0/FedKTL>



# Thanks!