Knowledge Distillation for Federated Learning: a Practical Guide

Alessio Mora, Irene Tenison, Paolo Bellavista, Irina Rish

Report: Xiachong Feng

Authors

Alessio Mora
PhD Student
University of Bologna

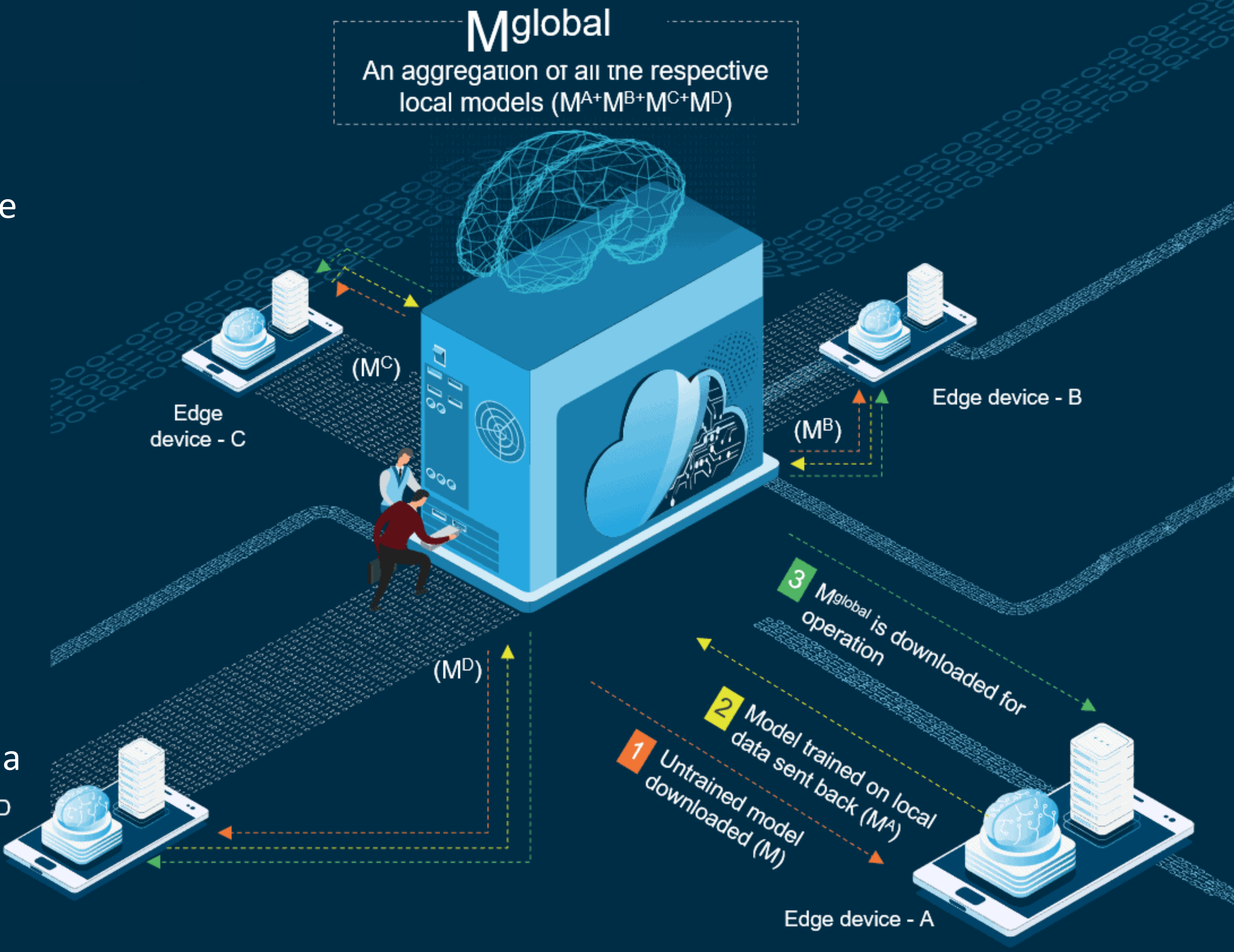
Irene Tenison
Student M.Sc.,
Université de Montréal

Paolo Bellavista
Full Professor
University of Bologna

Irina Rish
Full Professor
Université de Montréal,
Canada CIFAR AI Chair

Federated learning

- Federated learning (FL) provides a feasible solution to train a global model across multiple datasets *without raw data sharing*.
- FL is a *collaborative and privacy-aware learning paradigm*, which learns a global model by aggregating the models trained on local devices.
- Through FL, *each client would not worry about their private data exposed to other clients*, but they can collaboratively build a pre-trained model together.



Federated Averaging (FedAvg)

- The most used algorithms for FL are parameter-averaging based schemes (e.g., **Federated Averaging**)

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
  initialize w_0
  for each round $t = 1, 2, \dots$ **do**
    $m \leftarrow \max(C \cdot K, 1)$
    $S_t \leftarrow$ (random set of m clients)
    for each client $k \in S_t$ **in parallel do**
      $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \boxed{\frac{n_k}{n}} w_{t+1}^k$

ClientUpdate(k, w): // *Run on client* k
  $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
  for each local epoch i from 1 to E **do**
    for batch $b \in \mathcal{B}$ **do**
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return w to server

#participants

#samples of participant k

central model parameter

$$\boxed{w_{t+1}} \leftarrow \sum_{k=1}^{\boxed{K}} \frac{\boxed{n_k}}{\boxed{n}} \boxed{w_{t+1}^k}$$

local model parameter of participant k

#samples of all participants

Limits

- Clients must implement the same model architecture;
- Transmitting model weights and model updates implies high communication cost, which scales up with the number of model parameters;
- In presence of non-IID data distributions, parameter-averaging aggregation schemes perform poorly due to client model drifts.

#participants

#samples of participant k

central model parameter

$$\boxed{w_{t+1}} \leftarrow \sum_{k=1}^{\boxed{K}} \frac{\boxed{n_k}}{\boxed{n}} \boxed{w_{t+1}^k}$$

local model parameter of participant k

#samples of all participants

Knowledge Distillation

True label hot encoded

From teacher

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{CE}(q^S, y) + \lambda\mathcal{L}_{KL}(q^S_\tau, q^T_\tau)$$

Cross-entropy loss

Kullback-Leibler (KL) divergence

Knowledge Distillation (KD) For Fed

Overall

- Initially, KD-based strategies, also motivated by encouraging privacy properties, have been introduced to enable model heterogeneity and to reduce the communication cost of the process by exchanging model outputs and/or model-agnostic intermediate representations instead of directly transferring model parameters/model updates

For Server (server-side fusion)

- Then, a set of strategies proposed to enhance the aggregation step of FedAvg with a server-side ensemble distillation phase to enable model heterogeneity and/or improve model fusion in presence of heterogeneous.

For client (client model drift)

- Recently, two KD-based lines of work focused on mitigating the phenomenon of client model drift – which makes averaging-based aggregations inefficient – either using regularization terms in clients' objective functions or leveraging globally learned data-free generator.

Structure of This paper

According to the purpose KD is used for

- Model-agnostic FL
  - FL algorithms that use KD to enable model heterogeneity
    - Solutions that leverage server-side ensemble distillation on top of FedAvg's aggregation phase.
    - Communication-efficient strategies that enable model heterogeneity via exchanging locally-computed statistics, model outputs and/or model-agnostic intermediate features instead of model parameters.
- Non-IID data
  - FL algorithms that use KD to mitigate the impact of data heterogeneity on global model performance.
    - Server-side strategies that refine FedAvg's aggregation with a distillation phase
    - Client-side techniques that locally distill global knowledge to directly tackle client drift

Model-agnostic FL via KD

According to the purpose KD is used for

Model-agnostic FL

FL algorithms that use KD to enable model heterogeneity

Solutions that leverage server-side ensemble distillation on top of FedAvg's aggregation phase.

Communication-efficient strategies that enable model heterogeneity via exchanging locally-computed statistics, model outputs and/or model-agnostic intermediate features instead of model parameters.

Non-IID data

FL algorithms that use KD to mitigate the impact of data heterogeneity on global model performance.

Client-side techniques that locally distill global knowledge to directly tackle client drift

Server-side ensemble distillation

- FedAvg's protocol can be enhanced to enable model heterogeneity by leveraging server-side ensemble distillation on top of the aggregation step

The server can maintain a set of prototypical models, with each prototype representing all learners with same architecture. After collecting updates from clients, the server firstly performs a per-prototype aggregation and then produces soft targets for each received client model either leveraging unlabeled data or synthetically generated examples.

Next, such soft targets are averaged and used to fine tune each aggregated model prototype, exchanging knowledge among clients with different model architecture.

- [30] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. Advances in Neural Information Processing Systems, 33:2351–2363, 2020.

- [41] Felix Sattler, Tim Korjakow, Roman Rischke, and Wojciech Samek. Fedaux: Leveraging unlabeled auxiliary data in federated learning. IEEE Transactions on Neural Networks and Learning Systems, 2021.

Ensemble distillation for robust model fusion in federated learning

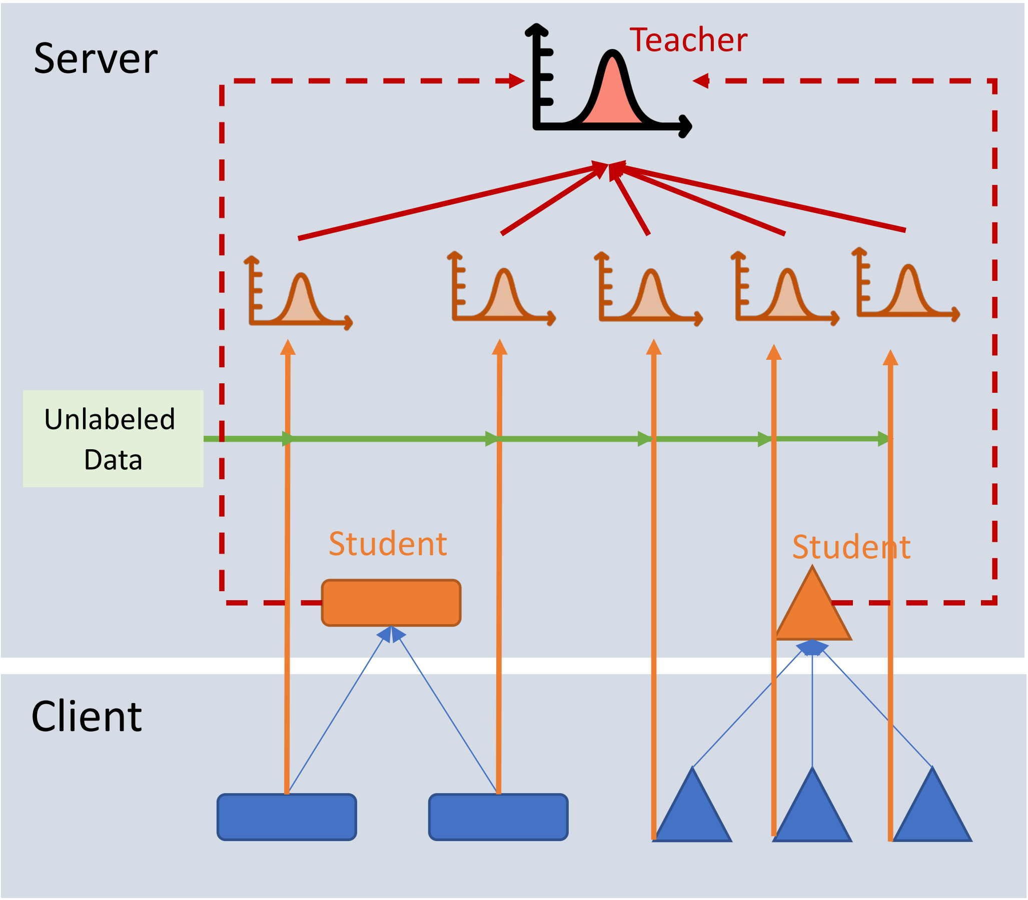
NeurIPS 2020

Algorithm 3 Illustration of FedDF for heterogeneous FL systems. The K clients are indexed by k, and n_k indicates the number of data points for the k-th client. The number of communication rounds is T, and C controls the client participation ratio per communication round. The number of total iterations used for model fusion is denoted as N. The distinct model prototype set $\mathcal{P}$ has p model prototypes, with each initialized as $\mathbf{x}_0^P$.

procedure SERVER
- initialize HashMap $\mathcal{M}$: map each model prototype P to its weights $\mathbf{x}_0^P$.
- initialize HashMap $\mathcal{C}$: map each client to its model prototype.
- initialize HashMap $\tilde{\mathcal{C}}$: map each model prototype to the associated clients.
- **for** each communication round $t = 1, \dots, T$ **do**
  - $\mathcal{S}_t \leftarrow$ a random subset (C fraction) of the K clients
  - **for** each client $k \in \mathcal{S}_t$ **in parallel do**
    - $\hat{\mathbf{x}}_t^k \leftarrow$ Client-LocalUpdate(k, $\mathcal{M}[\mathcal{C}[k]]$) ▷ detailed in Algorithm 2.
  - **for** each prototype $P \in \mathcal{P}$ **in parallel do**
    - initialize the client set $\mathcal{S}_t^P$ with model prototype P, where $\mathcal{S}_t^P \leftarrow \tilde{\mathcal{C}}[P] \cap \mathcal{S}_t$
    - initialize for model fusion $\mathbf{x}_{t,0}^P \leftarrow \sum_{k \in \mathcal{S}_t^P} \frac{n_k}{\sum_{k \in \mathcal{S}_t^P} n_k} \hat{\mathbf{x}}_t^k$
    - **for** j in $\{1, \dots, N\}$ **do**
      - sample $\mathbf{d}$, from e.g. (1) an unlabeled dataset, (2) a generator
      - use ensemble of $\{\hat{\mathbf{x}}_t^k\}_{k \in \mathcal{S}_t}$ to update server student $\mathbf{x}_{t,j}^P$ through AVGLOGITS
    - $\mathcal{M}[P] \leftarrow \mathbf{x}_{t,N}^P$
- **return** $\mathcal{M}$

KD

Enable aggregation from heterogeneous systems



Structure of This paper

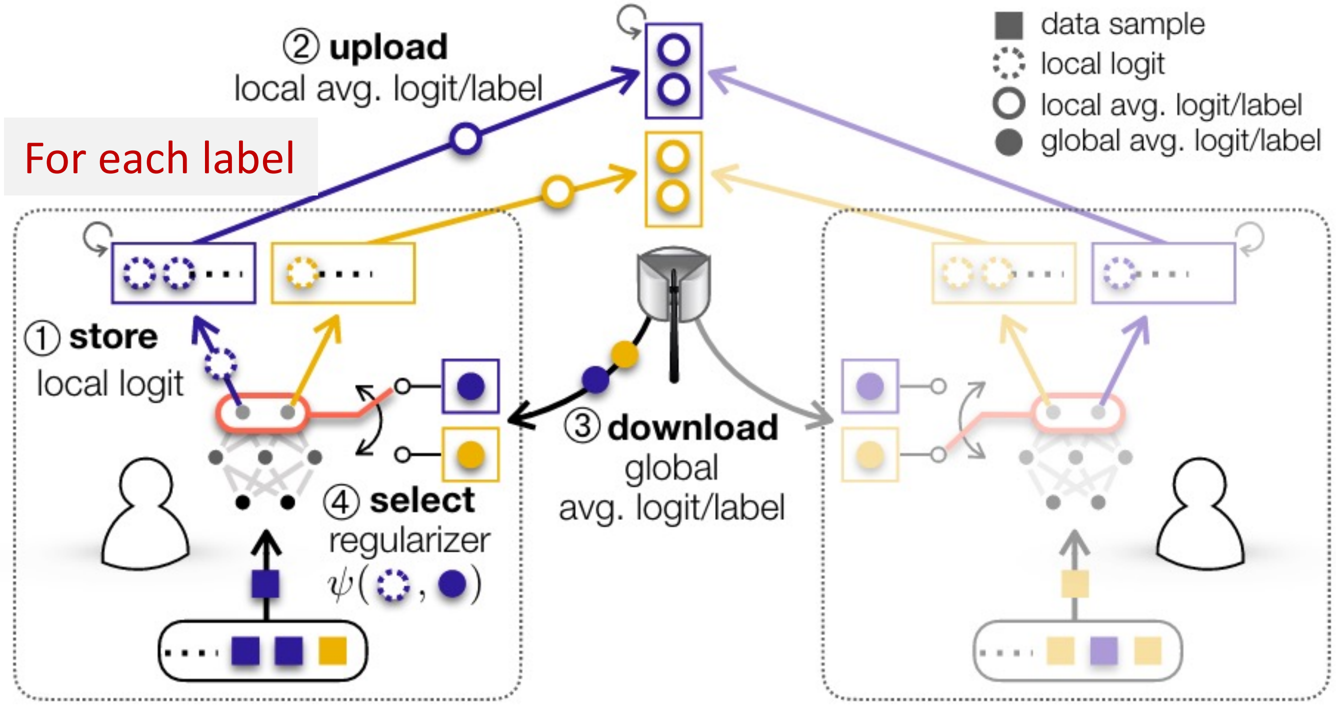
According to the purpose KD is used for

- Model-agnostic FL
  - FL algorithms that use KD to enable model heterogeneity
    - Disclosing aggregated statistics of **model outputs** on local data.
    - Exchanging model responses on proxy data
    - Leveraging intermediate features.
- Non-IID data
  - FL algorithms that use KD to mitigate the impact of data heterogeneity on global model performance.
    - Solutions that leverage server-side ensemble distillation on top of FedAvg's aggregation phase.
    - Server-side strategies that refine FedAvg's aggregation with a distillation phase
    - Client-side techniques that locally distill global knowledge to directly tackle client drift

Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data.

NIPS 2018 2nd Workshop on Machine Learning on the Phone and other Consumer Devices (MLPCD 2)

Core: exchange outputs



(a) FD with 2 devices and 2 labels.

Structure of This paper

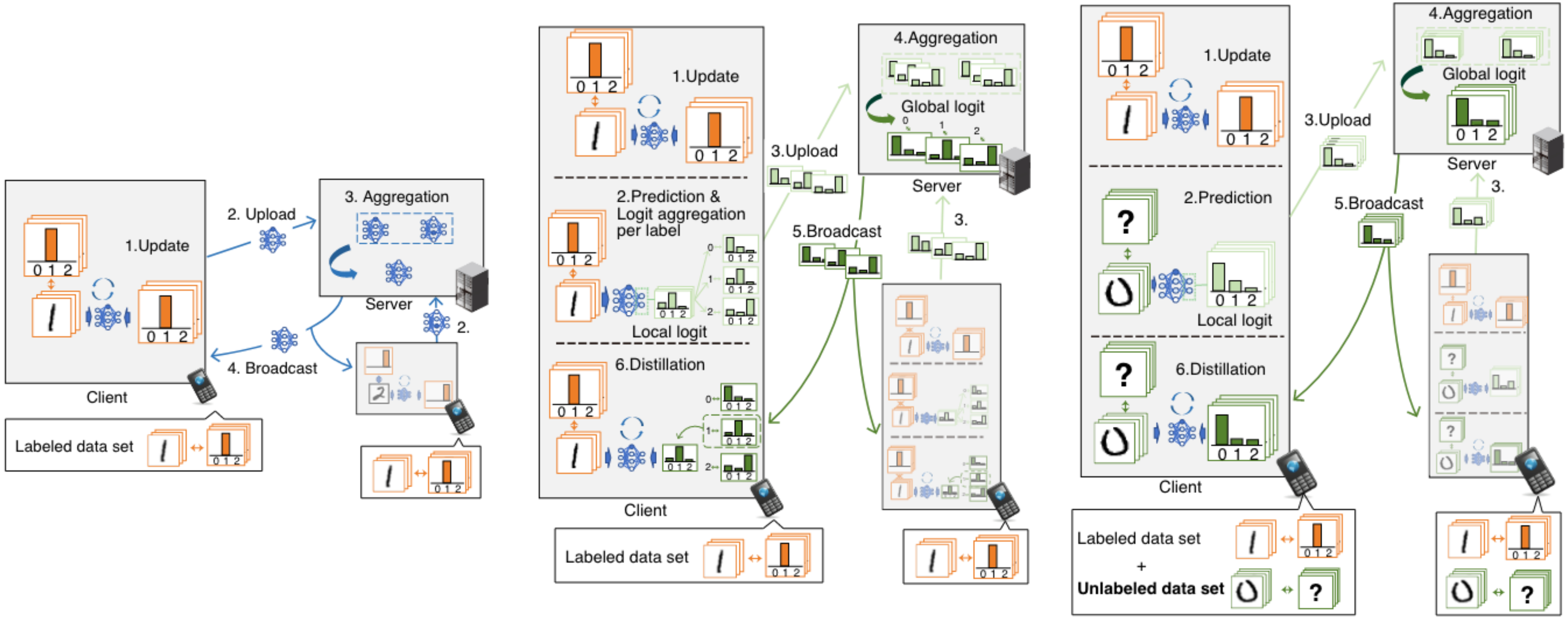
According to the purpose KD is used for

- Model-agnostic FL
  - FL algorithms that use KD to enable model heterogeneity
    - Solutions that leverage server-side ensemble distillation on top of FedAvg's aggregation phase.
    - Disclosing aggregated statistics of model outputs on local data.
    - Exchanging model responses on **proxy data**
    - Leveraging intermediate features.
- Non-IID data
  - FL algorithms that use KD to mitigate the impact of data heterogeneity on global model performance.
    - Server-side strategies that refine FedAvg's aggregation with a distillation phase
    - Client-side techniques that locally distill global knowledge to directly tackle client drift

Exchanging model responses on proxy data

1. **Broadcast:** clients receive the current global logits/soft targets;
2. **Local distillation:** clients distill their local model by mimicking the received global logits/soft-labels on a subset of the proxy dataset;
3. **Local training:** clients fine-tune the distilled model on local data;
4. **Local prediction:** clients compute their local logits/soft targets on the proxy dataset;
5. **Aggregation:** the server collects the logits/soft targets and aggregates them to produce the updated global logits/soft targets.

Distillation-Based Semi-Supervised Federated Learning for Communication-Efficient Collaborative Training with Non-IID Private Data



(a) Benchmark 1: Federated Learning with model parameter exchange [4].

(b) Benchmark 2: Federated Distillation [6].

(c) Proposed: **Distillaion-Based Semi-Supervised Fedetrated Learning**.

Fig. 1. Operational structures for benchmark schemes and proposed DS-FL.

Structure of This paper

According to the purpose KD is used for

Model-agnostic FL

FL algorithms that use KD to enable model heterogeneity

- Disclosing aggregated statistics of model outputs on local data.
- Exchanging model responses on proxy data
- **Leveraging intermediate features.**

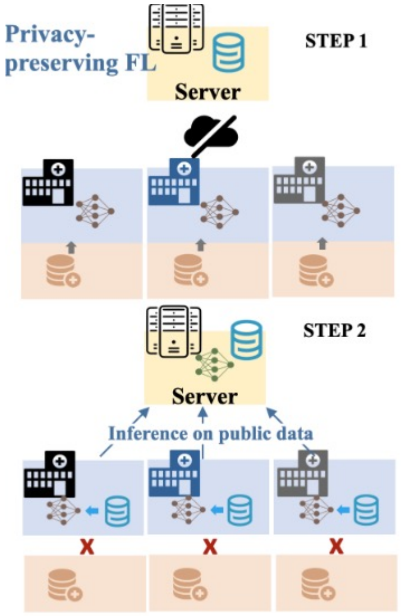
Non-IID data

FL algorithms that use KD to mitigate the impact of data heterogeneity on global model performance.

- Solutions that leverage server-side ensemble distillation on top of FedAvg's aggregation phase.
- Server-side strategies that refine FedAvg's aggregation with a distillation phase
- Client-side techniques that locally distill global knowledge to directly tackle client drift

Ensemble Attention Distillation for Privacy-Preserving Federated Learning

ICCV 2021



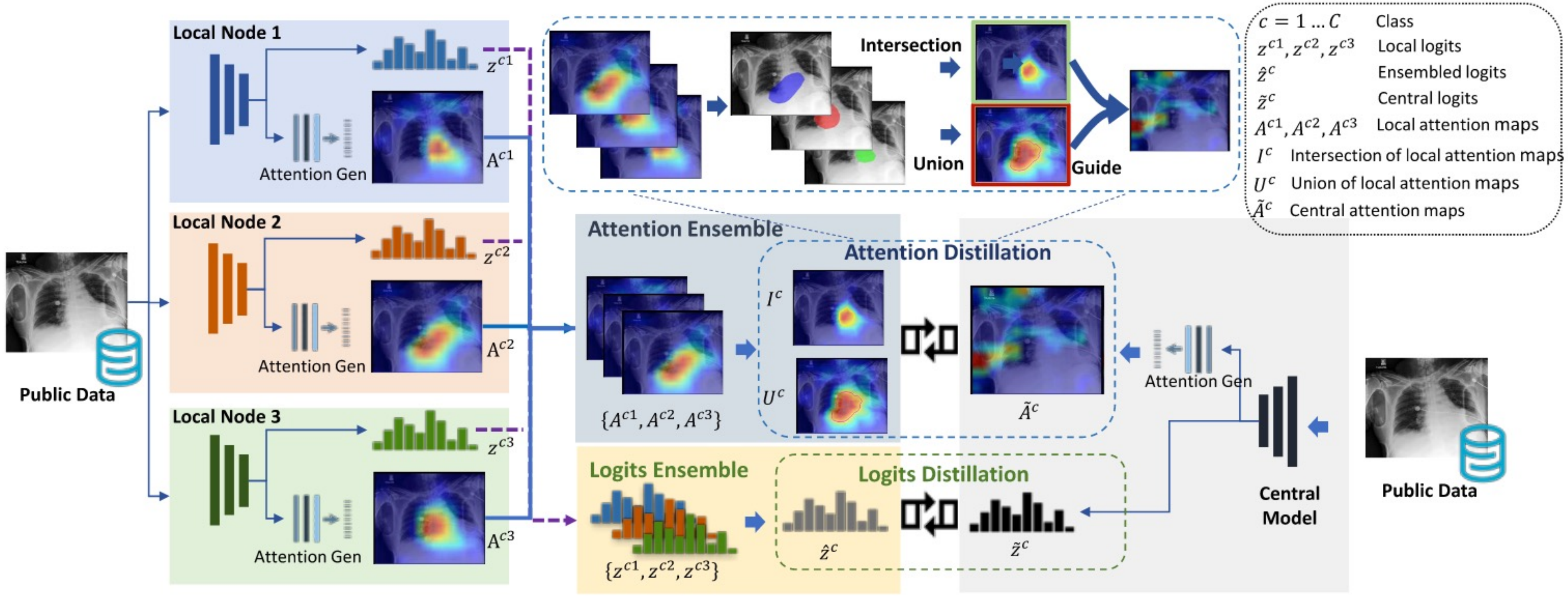
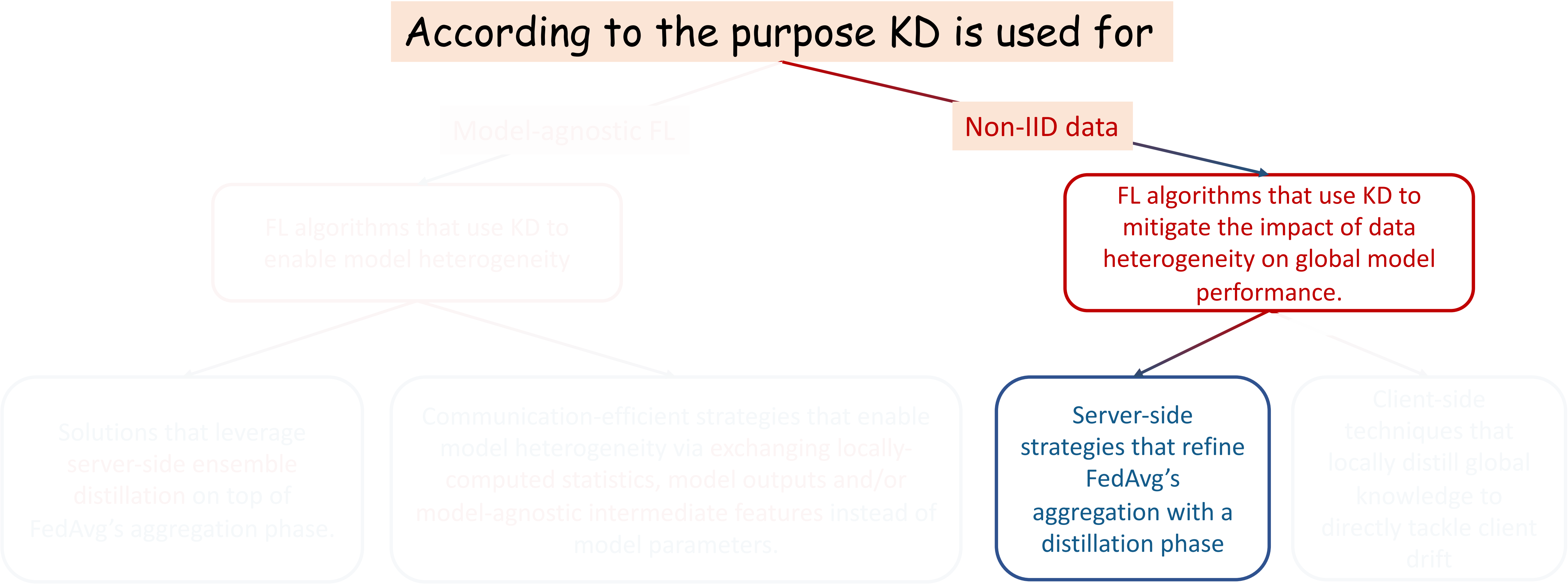


Figure 2. **Overview of the proposed FedAD framework.**

Server-side KD-based refinement of global model



Ensemble distillation for robust model fusion in federated learning

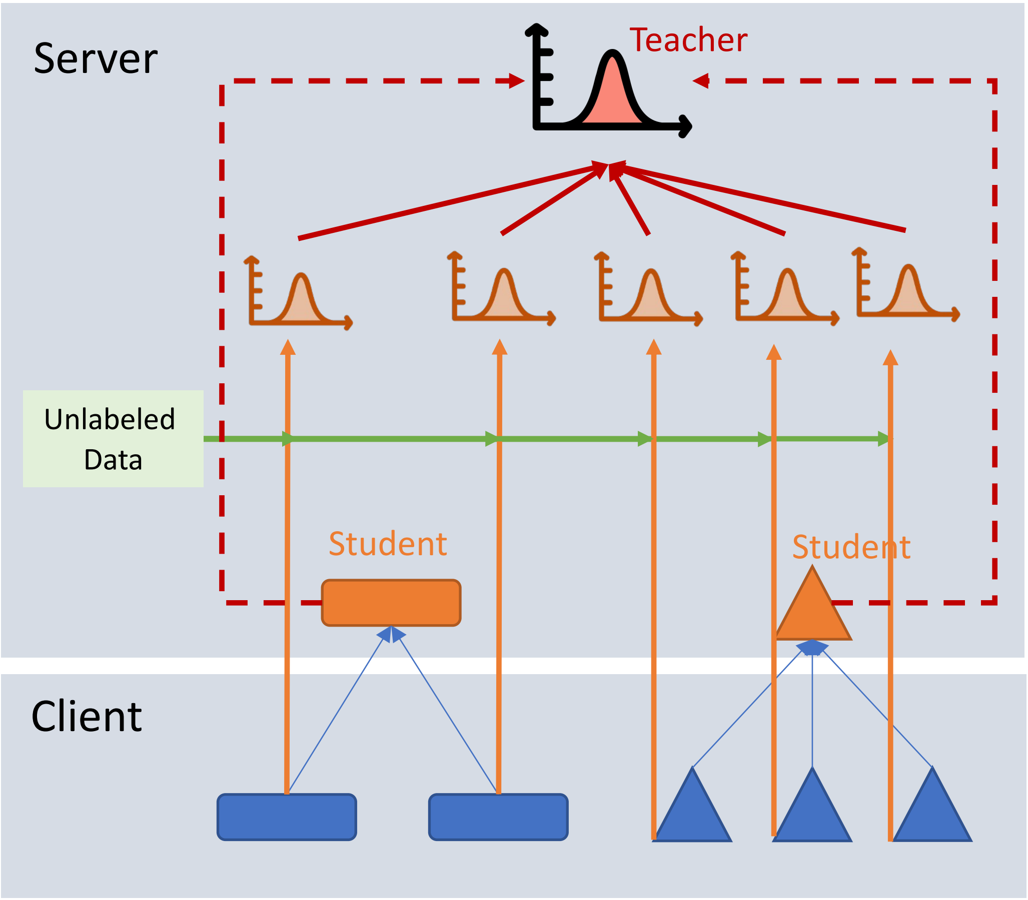
NeurIPS 2020

Algorithm 3 Illustration of FedDF for heterogeneous FL systems. The K clients are indexed by k, and n_k indicates the number of data points for the k-th client. The number of communication rounds is T, and C controls the client participation ratio per communication round. The number of total iterations used for model fusion is denoted as N. The distinct model prototype set $\mathcal{P}$ has p model prototypes, with each initialized as $\mathbf{x}_0^P$.

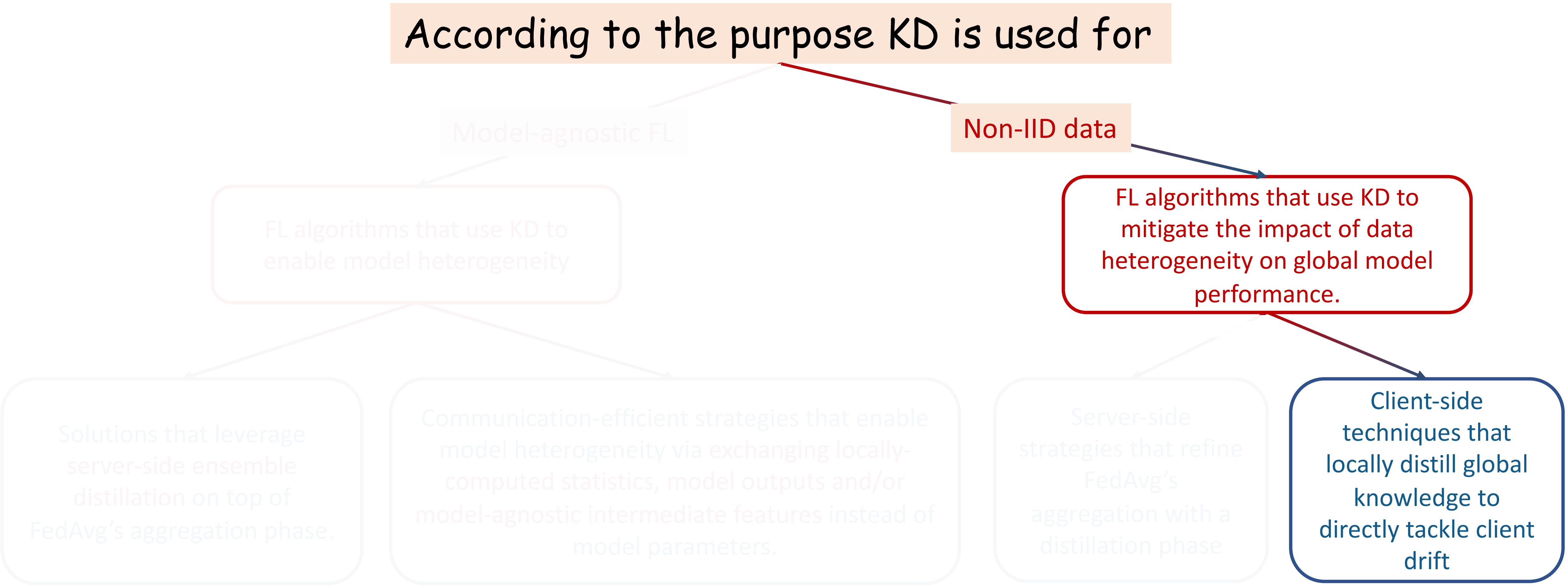
procedure SERVER
  initialize HashMap $\mathcal{M}$: map each model prototype P to its weights $\mathbf{x}_0^P$.
  initialize HashMap $\mathcal{C}$: map each client to its model prototype.
  initialize HashMap $\tilde{\mathcal{C}}$: map each model prototype to the associated clients.
  for each communication round $t = 1, \dots, T$ **do**
    $\mathcal{S}_t \leftarrow$ a random subset (C fraction) of the K clients
    for each client $k \in \mathcal{S}_t$ **in parallel do**
      $\hat{\mathbf{x}}_t^k \leftarrow$ Client-LocalUpdate$(k, \mathcal{M}[\mathcal{C}[k]])$ ▷ detailed in Algorithm 2.
    for each prototype $P \in \mathcal{P}$ **in parallel do**
      initialize the client set $\mathcal{S}_t^P$ with model prototype P, where $\mathcal{S}_t^P \leftarrow \tilde{\mathcal{C}}[P] \cap \mathcal{S}_t$
      initialize for model fusion $\mathbf{x}_{t,0}^P \leftarrow \sum_{k \in \mathcal{S}_t^P} \frac{n_k}{\sum_{k \in \mathcal{S}_t^P} n_k} \hat{\mathbf{x}}_t^k$
      for j in $\{1, \dots, N\}$ **do**
        sample $\mathbf{d}$, from e.g. (1) an unlabeled dataset, (2) a generator
        use ensemble of $\{\hat{\mathbf{x}}_t^k\}_{k \in \mathcal{S}_t}$ to update server student $\mathbf{x}_{t,j}^P$ through AVGLOGITS
      $\mathcal{M}[P] \leftarrow \mathbf{x}_{t,N}^P$
  return $\mathcal{M}$

KD

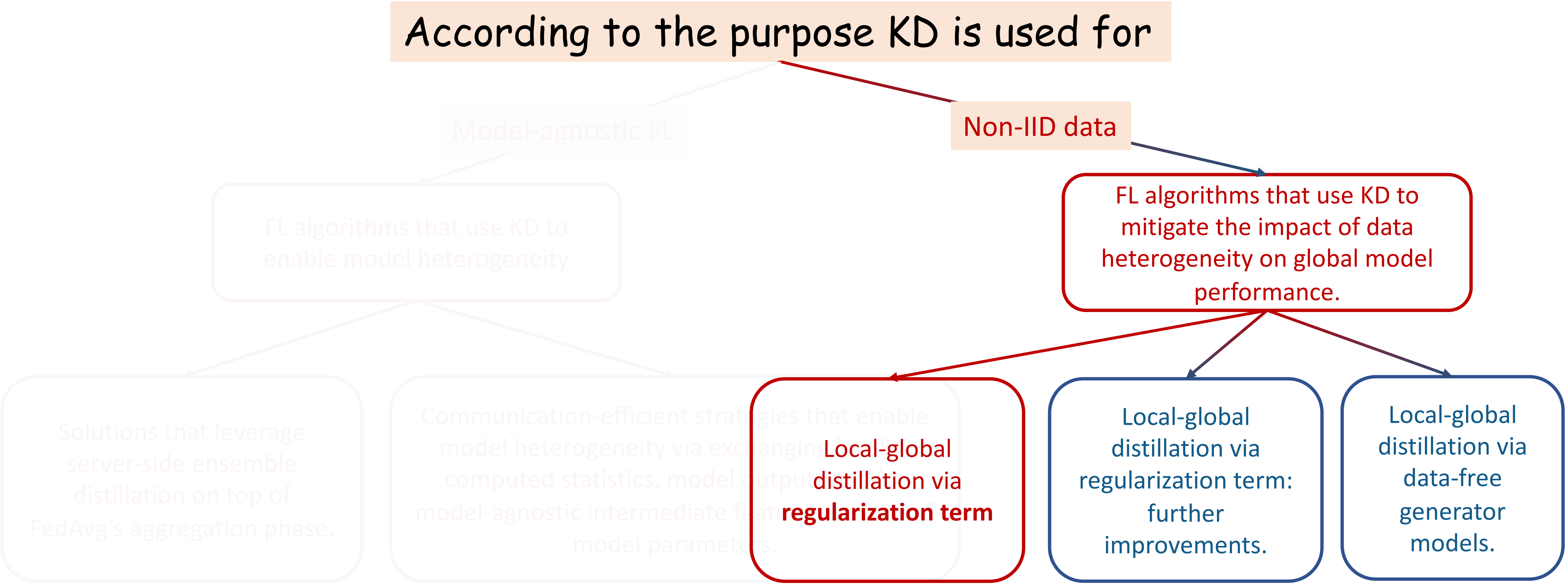
Enable aggregation from heterogeneous systems



Local distillation of global knowledge

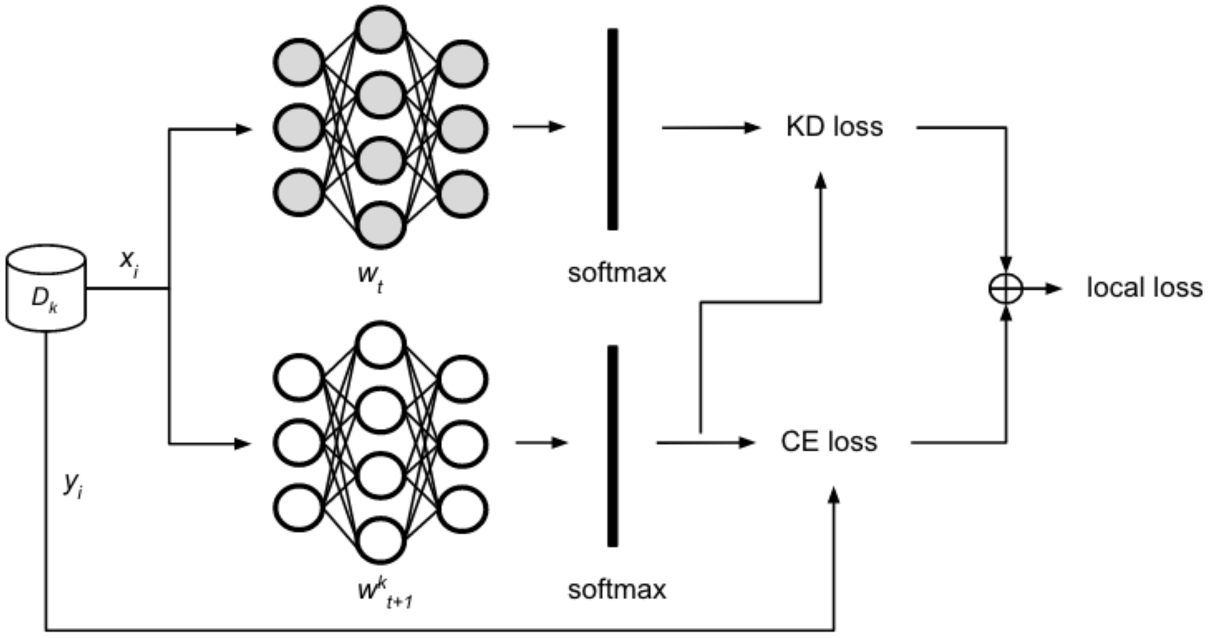


Local distillation of global knowledge



Local-global distillation via regularization term

- In local-global distillation, the local objective function of clients becomes a **linear combination** between the **cross-entropy loss** and a **KD-based loss** that measures the discrepancy among the global model's output (i.e., the teacher model's output) and the local model's output (i.e., the student model output) on private data, e.g. via Kullback-Leibler divergence.



Local distillation of global knowledge

According to the purpose KD is used for

- Model-agnostic FL
  - FL algorithms that use KD to enable model heterogeneity
    - Solutions that leverage server-side ensemble distillation on top of FedAvg's aggregation phase.
- Non-IID data
  - FL algorithms that use KD to mitigate the impact of data heterogeneity on global model performance.
    - Local-global distillation via regularization term
    - Local-global distillation via **regularization term: further improvements**
    - Local-global distillation via data-free generator models

Learning Critically: Selective Self Distillation in Federated Learning on Non-IID Data

IEEE Transactions on Big Data 2022

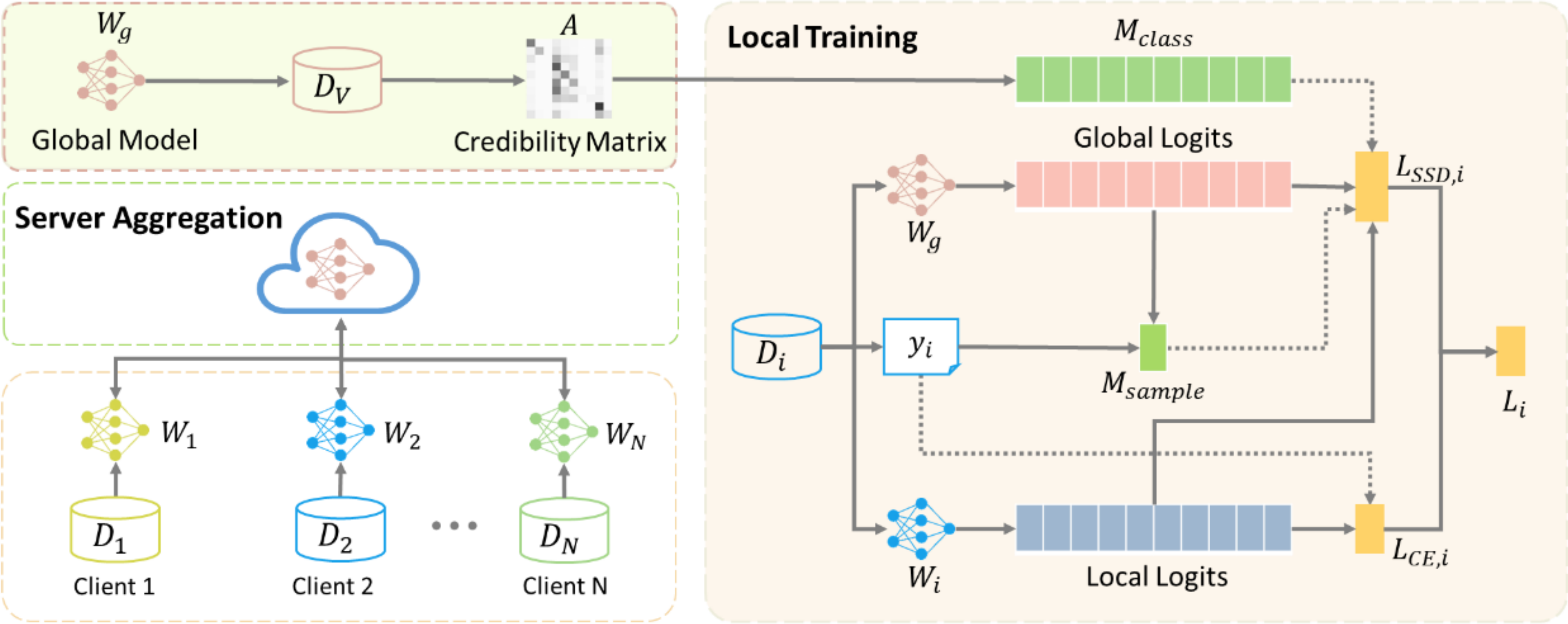


Fig. 5: An overview of FedSSD in the heterogeneous setting.

Local distillation of global knowledge

According to the purpose KD is used for

Model-agnostic FL

FL algorithms that use KD to enable model heterogeneity

Solutions that leverage server-side ensemble distillation on top of FedAvg's aggregation phase.

Communication-efficient strategies that enable model heterogeneity via exchanging locally-computed statistics, model outputs and/or model-agnostic intermediate features instead of model parameters.

Non-IID data

FL algorithms that use KD to mitigate the impact of data heterogeneity on global model performance.

Local-global distillation via regularization term

Local-global distillation via regularization term: further improvements

Local-global distillation via data-free generator models

Data-Free Knowledge Distillation for Heterogeneous Federated Learning
PMLR 2021

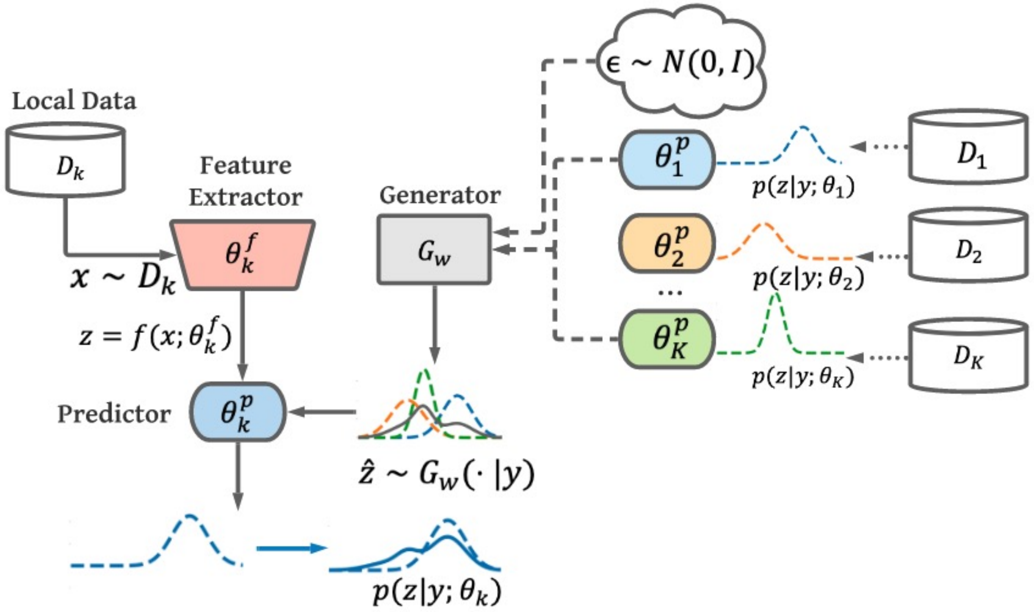


Figure 1. Overview of FEDGEN: a generator $G_w(\cdot|y)$ is learned by the server to aggregate information from different local clients without observing their data. The generator is then sent to local users, whose knowledge is distilled to user models to adjust their interpretations of a good feature distribution.

Thanks~