

Knowledge Distillation in Federated Learning: A Practical Guide

Alessio Mora¹, Irene Tenison², Paolo Bellavista¹ and Irina Rish³

¹University of Bologna

²Massachusetts Institute of Technology (MIT)

³Mila, University of Montreal

alessio.mora@unibo.it, itenison@mit.edu, paolo.bellavista@unibo.it, irina.rish@mila.quebec

Abstract

Federated Learning (FL) enables the training of Deep Learning models without centrally collecting possibly sensitive raw data. The most used algorithms for FL are parameter-averaging based schemes (e.g., Federated Averaging) that, however, have well known limits, i.e., model homogeneity, high communication cost, poor performance in presence of heterogeneous data distributions. Federated adaptations of regular Knowledge Distillation (KD) can solve or mitigate the weaknesses of parameter-averaging FL algorithms while possibly introducing other trade-offs. In this article, we originally present a focused review of the state-of-the-art KD-based algorithms specifically tailored for FL, by providing both a novel classification of the existing approaches and a detailed technical description of their pros, cons, and tradeoffs.

1 Introduction

Federated Learning (FL) has been proposed as an alternative to cloud-based Deep Learning (DL). This paradigm decouples the ability to train DL models from the need of harvesting raw data, alternating on-device computation and periodic communication [McMahan *et al.*, 2016; Bellavista *et al.*, 2021]. During the learning process, only ephemeral and locally processed payloads need to be disclosed by the participants in the federation, making it harder to infer private information about the individuals.

Federated Averaging (FedAvg) represents the baseline algorithm for Federated Learning (FL) [McMahan *et al.*, 2016]. In FedAvg, collaborative learning proceeds in synchronous rounds by leveraging a client-server paradigm. Participants (i.e., clients) iteratively exchange model updates and model weights with a central aggregator (i.e., the server) to collaboratively build a global model. Round by round, the server aggregates model updates by weighted average and distributes the new version of the global model.

However, parameter-averaging aggregation schemes, such as FedAvg, have well-known limits. Firstly, this class of algorithms implies model homogeneity among the federation, i.e., each client is constrained to use the same neural architecture since the server directly merges clients' updates (e.g.,

by weighted average). This may be an issue when the federation of learners consists of clients with heterogeneous hardware capabilities. Furthermore, exchanging model parameters and model updates have high communication cost, which scales with the number of model parameters – even though a plethora of strategies (e.g., [Sattler *et al.*, 2019]) have been proposed to improve the communication efficiency at the cost of global model performance. In addition, exchanging model parameters/updates exposes client to information leakage, and the server must know the architecture and structure of clients' model to broadcast the global parameters, possibly incurring in intellectual property issues (i.e., clients in the federation are unwilling to share the architecture they are using). Lastly, but not less important, when clients hold heterogeneous data, local models tend to diverge from each other during training and fine tune on private examples (i.e., client drift). As a consequence, directly aggregating model parameters/updates degrades the global model performance [Karimreddy *et al.*, 2020].

This article focuses on specifically reviewing the FL-oriented adaptations of regular Knowledge Distillation (KD) techniques that have been employed to alleviate the above mentioned weaknesses of FL parameter-averaging aggregation schemes. Initially, KD-based strategies, also motivated by encouraging privacy properties [Papernot *et al.*, 2016], have been introduced to enable model heterogeneity and to reduce communication costs by exchanging model outputs and/or model-agnostic intermediate representations instead of directly transferring model parameters/updates. Then, a set of strategies have been 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 data. Recently, two KD-based lines of work have 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.

To our knowledge, this is the first paper that provides a systematic categorization of KD-based methods tailored to address specific FL issues. The paper mainly makes the following contributions:

- We propose a novel taxonomy of KD-based methods for FL, which can help researchers to better understand the

potential and possible further applications of distillation-inspired methods.

- We present a detailed technical overview of the existing KD-based methods, of their primary scope, and of the rationale behind their design/implementation choices. In addition, we discuss their strengths and possible drawbacks.
- We outline the emerging use of KD in this domain, by highlighting promising directions for future research.

The paper is organized as follows. Section 2 introduces the fundamental of KD. Section 3 presents FL algorithms that use KD to enable model heterogeneity. Section 4 describes FL algorithms that use KD to mitigate the impact of data heterogeneity on global model performance. Finally, in Section 5, we concisely highlight the emerging use of KD for other relevant FL issues.

2 Background

2.1 Knowledge Distillation

Knowledge Distillation (KD) methods have been designed to transfer knowledge from a larger deep neural network, the *teacher*, to a lightweight network, the *student* [Hinton *et al.*, 2015]. In the simplest form of KD, the student model learns by mimicking the (pre-trained) teacher model’s outputs on a proxy dataset, also called transfer set. If the transfer set is labeled, the student can be trained using a linear combination of two loss functions,

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{CE}(\tilde{y}^S, y) + \lambda\tau^2\mathcal{L}_{KD}(\tilde{y}^S, \tilde{y}^T) \quad (1)$$

\mathcal{L}_{CE} is the usual cross-entropy loss between the true label y (e.g., hot encoded) and the class probabilities \tilde{y}^S (i.e., soft targets) predicted by the student neural network. Soft targets \tilde{y}_τ are typically produced by applying a softmax layer to the logits z_i so that $\tilde{y}(i) = \frac{\exp(z_i/\tau)}{\sum_j \exp(z_j/\tau)}$, where z_i is the i -th value of logits vector z . The temperature τ controls the softness of the probability distribution. \tilde{y}^S is computed with τ set to 1. λ weights the impact of the two loss terms. \mathcal{L}_{KD} is a general distillation loss that measures the distance between the distribution of student’s soft targets (\tilde{y}^S) and the the distribution of teacher’s soft targets (\tilde{y}^T), e.g. via Kullback-Leibler (KL) divergence. Alternatively, \mathcal{L}_{KD} could directly measure the error between student and teacher logits (e.g., mean squared error). We refer to [Gou *et al.*, 2021] for taxonomy and recent progress in the KD area.

2.2 Codistillation

Codistillation (CD) refers to an online version of distillation, which obviates the need of a pre-trained teacher in regular KD [Anil *et al.*, 2018; Gou *et al.*, 2021]. In fact, CD simultaneously trains T copies of a model by adding a distillation term to the regular loss function of the j th model to mimic the average prediction of the other $T - 1$ models. In this way, each worker network sees the ensemble of the other models as a virtual teacher. For CD, the pre-trained teacher’s soft targets in Eq. 1 is replaced with the ensemble soft targets of $T - 1$ workers.

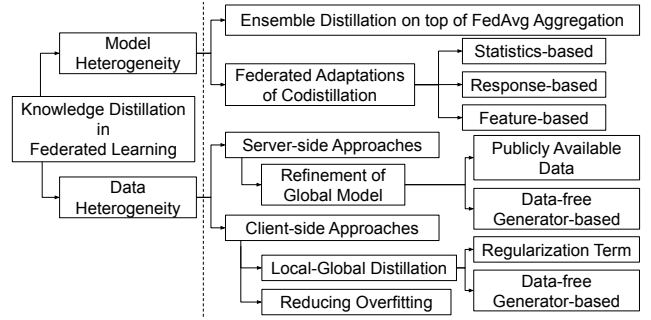


Figure 1: Taxonomy of KD-based solutions for FL issues.

In the original formulation of CD [Anil *et al.*, 2018], (1) all the workers implement the same neural architecture, (2) all the workers use the same dataset for training and, most notably, (3) the distillation loss is employed during training before any model has fully converged.

2.3 Proposed Taxonomy and Classification

Figure 1 illustrates our taxonomy of the possible KD-based mechanisms designed to enable model heterogeneity or to combat the effects of data heterogeneity. Table 1 lists the most relevant related work in the literature and surveyed in this paper, by classifying the proposed solutions according to their primary aim. For each solution, we detail the kind of per-round exchanged information, the need of auxiliary data, and the type of KD involved. For the latter, a regularizer-based approach uses KD to regularize local training. Generator-based mechanism leverages a generator model to assemble synthetic data and transfer knowledge as inductive biases. Digestion means that knowledge is absorbed by imitating teacher outputs on the same proxy data.

3 Enabling FL Model Heterogeneity via KD

KD has been initially designed to transfer knowledge among neural networks with different structure and depth. In this Subsection, we review strategies that adopt KD to enable model agnosticism in FL, i.e., transfer knowledge among clients with heterogeneous model architectures.

3.1 Enhancing FedAvg Aggregation

FedAvg’s protocol can be enhanced to enable model heterogeneity by leveraging server-side ensemble distillation on top of the aggregation step [Lin *et al.*, 2020; Sattler *et al.*, 2021a], through which knowledge is transferred among clients with different model architecture. To this end, 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 model 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. Alternative possible solutions to enable model heterogeneity consist in exploiting distributed adaptations of CD instead of parameter-averaging algorithms such as FedAvg, as presented in the following.

	Purpose	Exchanged information		Auxiliary data	KD approach	
		Upload	Download		Client-side	Server-side
FedDistill [Jeong <i>et al.</i> , 2018]	MH, CE	\bar{z}	\bar{z}	data-free	regularizer	-
FedMD [Li and Wang, 2019]	MH, CE	\tilde{y}_p	\tilde{y}_p	labeled	digestion	-
Cronus [Chang <i>et al.</i> , 2019]	MH, CE	\tilde{y}_p	\tilde{y}_p	unlabeled	digestion	-
DS-FL [Itahara <i>et al.</i> , 2021]	MH, CE	\tilde{y}_p	\tilde{y}_p	unlabeled	digestion	-
MATH [Hu <i>et al.</i> , 2021]	MH, CE	\tilde{y}_p	\tilde{y}_p	labeled	digestion	digestion
CFD [Sattler <i>et al.</i> , 2021b]	MH, CE	\tilde{y}_p	\tilde{y}_p	unlabeled	digestion	digestion
FedGEMS [Cheng <i>et al.</i> , 2021]	MH, CE	\tilde{y}_p	\tilde{y}_p	labeled	digestion	digestion
FedAD [Gong <i>et al.</i> , 2021]	MH, CE	z_p, A_p	-	unlabeled	-	digestion
FedGKT [He <i>et al.</i> , 2020]	MH, CE	z, H, y	z	data-free	regularizer	regularizer
FedDKC [Wu <i>et al.</i> , 2022b]	MH, CE	z, H, y	z	data-free	regularizer	regularizer
FedDF [Lin <i>et al.</i> , 2020]	MH, NIID	w	w	unlabeled	-	digestion
FedAUX [Sattler <i>et al.</i> , 2021a]	MH, NIID	w	w	unlabeled	-	digestion
FedBE [Chen and Chao, 2020]	NIID	w	w	unlabeled	-	digestion
FedFTG [Zhang <i>et al.</i> , 2022b]	NIID	w, c	w	data-free	-	generator + digestion
FedZKT [Zhang <i>et al.</i> , 2022a]	NIID	w	w	data-free	-	generator + digestion
DaFKD [Wang <i>et al.</i> , 2023]	NIID	w, θ^d, θ^g	w, θ^g	data-free	generator	generator + digestion
FedGKD [Yao <i>et al.</i> , 2023]	NIID	w	w, w^h	data-free	regularizer	-
FedNTD [Lee <i>et al.</i> , 2022]	NIID	w	w	data-free	regularizer	-
FedLMD [Lu <i>et al.</i> , 2023]	NIID	w	w	data-free	regularizer	-
FedED [Guo <i>et al.</i> , 2024]	NIID	w	w	data-free	regularizer	-
FedCAD [He <i>et al.</i> , 2022b]	NIID	w	w, α_y	labeled	regularizer	-
FedSSD [He <i>et al.</i> , 2022a]	NIID	w	w, C	labeled	regularizer	-
FedMLB [Kim <i>et al.</i> , 2022]	NIID	w	w	data-free	regularizer	-
FedBR [Liu <i>et al.</i> , 2023b]	NIID, CE	w	w	data-free	regularizer	-
FedAlign [Mendieta <i>et al.</i> , 2022]	NIID	w	w	data-free	regularizer	-
FedDistill ⁺ [Yao <i>et al.</i> , 2023]	NIID	w, \bar{z}	w, \bar{z}	data-free	regularizer	-
FedGen [Zhu <i>et al.</i> , 2021]	NIID	w, c	w	data-free	generator	-
pFedSD [Jin <i>et al.</i> , 2022]	P	w	w	data-free	regularizer	-
Chen <i>et al.</i> [Chen <i>et al.</i> , 2023]	P	w	w	data-free	regularizer	-
Wu <i>et al.</i> [Wu <i>et al.</i> , 2022a]	FU	w	w	unlabeled	-	digestion
FedET [Liu <i>et al.</i> , 2023a]	CI, NIID	w, c	w	data-free	generator	generator + digestion
FedNASD [Wu <i>et al.</i> , 2024]	CI, NIID	w	w	data-free	regularizer	-

Table 1: Synoptic overview of the surveyed solutions. We have identified 6 possible categories for primary purpose: model heterogeneity (MH), non-iidness (NIID), communication efficiency (CE), unlearning (FU), personalization (P), class-incremental learning (CI). Upload refers to the client-to-server link. Symbols: w model parameters, z logit vectors (model output before softmax), \tilde{y}_p soft targets (model output after softmax) on public data, w^h historical model parameters, \bar{z} per-label average logit vectors, y labels of local data, H intermediate feature maps, A attention maps, α_y per-class adaptive weights, C local label count, c local label count, θ^g generator model weights, θ^d discriminator model weights. We do not differentiate among model weights and model updates.

3.2 Federated Adaptations of Codistillation

In a general federated adaptation of CD [Anil *et al.*, 2018], each client at round t acts as student and sees the ensemble of clients knowledge at round $t - 1$ as a virtual teacher knowledge. As highlighted in Section 2.2, traditional CD postulates that the worker networks have access to the same training dataset to form an ensemble of model responses on common samples. Such a requirement is not acceptable in the FL context, where collaborative training is performed without disclosing the private raw data of clients. Therefore, considering a classification task, federated adaptations of CD avoids the aforementioned problem of collecting model responses on common training data examples by exchanging a different kind of knowledge. Among the solutions in the literature, we identify three types of knowledge to enable federated versions of CD, and they are:

1. ensemble of aggregated statistics of model responses on

local data (e.g. per-label mean model responses),

2. ensemble of local model responses computed on a publicly available dataset and not on local data,
3. or ensemble of both model responses and model-agnostic intermediate features.

It is worth noting that the clients and the server exchange this kind of information instead of model parameters. Furthermore, FL adaptations of CD relax the constraint of implementing the same model architecture, which was needed in regular CD for datacenter-oriented distributed training. In fact, exchanging knowledge based on model responses (or on model-agnostic intermediate features) enables heterogeneous models among workers, i.e., FL clients, as long as they have the same output shape. Fig. 2 illustrates FL adaptations of CD with respect to FedAvg baseline. Table 2 sums up the comparison among FL adaptations of CD treated in the following subsections.

	Knowledge	Transfer Set	Server Model	Notes
FedDistill [Jeong <i>et al.</i> , 2018]	statistics	data-free	no	KD-based regularizer
FedMD [Li and Wang, 2019]	response	labeled	no	Pre-training on D_p
Cronus [Chang <i>et al.</i> , 2019]	response	unlabeled	no	Local training on $(X_p, \tilde{Y}_p) \cup D_k$
DS-FL [Itahara <i>et al.</i> , 2021]	response	unlabeled	no	Entropy Reduction Aggregation
MATH [Hu <i>et al.</i> , 2021]	response	labeled	yes	Server training on $(X_p, \tilde{Y}_p) \cup D_p$
CFD [Sattler <i>et al.</i> , 2021b]	response	unlabeled	yes	Compressed soft targets
FedGEMS [Cheng <i>et al.</i> , 2021]	response	labeled	yes	Server-side knowledge refinement
FedAD [Gong <i>et al.</i> , 2021]	feature	unlabeled	yes	One-shot algorithm
FedGKT [He <i>et al.</i> , 2020]	feature	data-free	yes	FL + Split learning paradigm
FedDKC [Wu <i>et al.</i> , 2022b]	feature	data-free	yes	Knowledge refinement

Table 2: Comparison among strategies to enable model heterogeneity via FL adaptations of CD. D_k represents the local private dataset of a generic client. D_p represents a public transfer set. (X_p, \tilde{Y}_p) is the public transfer dataset labeled with soft targets \tilde{Y}_p . Knowledge column is inspired by the classification in [Gou *et al.*, 2021]; statistics-based disclose aggregated statistics (e.g., per-label mean logit vector) of client model responses on local data, response-based methods communicate model outputs, feature-based also share intermediate representations.

Disclosing Aggregated Statistics of Model Responses on Local Data.

In [Jeong *et al.*, 2018], Jeong *et al.* presented a pioneering distillation-based baseline for FL, FedDistill, depicted in Fig. 2b. Participants periodically transmit only per-label mean soft targets computed on their private dataset. The server, in turn, averages such tensors and produces per-label global soft targets to be broadcast the next round. When locally training, clients regularize their local loss with a per-label distillation term which uses the received global soft targets as the teacher’s output. A similar strategy is later presented in [Seo *et al.*, 2020]. It is worth noting that FedDistill is extremely communication efficient with respect to parameter-based schemes when considering DL models for classification task, since the communication payload size depends on the model response size and not on model size.

Exchanging Model Responses on Publicly Available Data.

Federated CD can be enabled by using knowledge formed by an ensemble of model responses computed on a proxy transfer set (publicly available, both clients and server can retrieve it). In this way, clients train on their private data, and share knowledge via their model response on the transfer set. Here, the approaches in literature are more variegated, but a general skeleton of algorithmic steps can be the following (illustrated in Fig. 2c):

1. *broadcast*: clients receive the current global logits/soft targets;
2. *local digest*: clients distill their local model by mimicking the received global logits/soft-labels on a subset of the transfer dataset. Here, the parallel with traditional CD: each client sees the averaged predictions of the other clients at the previous round as a virtual teacher. This step can be also seen as a way to retrieve the global model parameters instead of directly receiving them from the server as in parameter-based schemes (e.g., FedAvg);
3. *revisit (local train)*: clients fine-tune the distilled model on local data;
4. *local predict*: clients compute their local logits/soft targets on a subset of the transfer dataset;

5. *upload*: clients sends back the computed logits/soft targets;
6. *aggregate*: the server aggregates the client predictions to produce the updated global logits/soft targets.
7. (*optional*) *global digest*: the server distills a model from the aggregated soft targets on the respective subset of the transfer dataset, and uses it again to generate the global logits/soft targets to broadcast. Learning a server-side model, that is refined round by round, can improve the training process when there is partial participation of clients. Next, a new round begins.

Table 2 summarizes the deviation of the surveyed works from the above general algorithmic steps. In particular, considering either a labeled or unlabeled proxy dataset influences the design of algorithms. FedMD [Li and Wang, 2019] uses a proxy labeled dataset to perform an initial pretraining phase on clients, before the protocol starts. Itahara *et al.* modify the aggregation step proposing an Entropy Reduction Aggregation (ERA), demonstrating that using a temperature lower than 1 when applying softmax to the aggregated logits reduces the entropy of global soft targets, and can help the training process, especially in non-IID settings [Itahara *et al.*, 2021]. Compressed Federated Distillation (CFD) [Sattler *et al.*, 2021b] implements an extreme and effective compression technique for soft targets based on quantization and delta coding, which is applied both by clients and server before communicating. Cronus [Chang *et al.*, 2019] merges the *local digest* and *revisit* step by directly training on the union (i.e., concatenation) of the private dataset and the soft-labeled public one. In addition, Cronus aggregates soft targets following the approach in [Diakonikolas *et al.*, 2017] for enhanced robustness. Similarly to Cronus, in MATH [Hu *et al.*, 2021] clients jointly train on private dataset, public dataset, and public dataset tagged with global soft targets. MATH [Hu *et al.*, 2021] considers a labeled proxy dataset, and distills its server model by training it on the union of such a public dataset with the soft-labeled version of it. FedGEM [Cheng *et al.*, 2021] proposes to employ a powerful model server, and its variation FedGEMS exploits the labels in the public transfer set to enforce a selection and weighting strategy which can improve the server-side digestion [Cheng *et al.*, 2021].

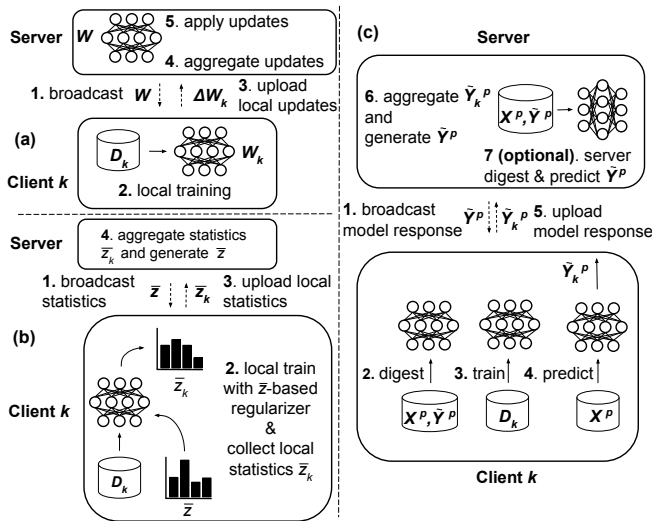


Figure 2: KD-based solutions for FL issues. (a) FedAvg, (b) statistic-based federated CD, (c) response-based federated CD.

Leveraging Intermediate Features. FedAD [Gong *et al.*, 2021] also uses intermediate features besides model output to extend response-based knowledge distillation. The intermediate features are model-agnostic attention maps [Selvaraju *et al.*, 2017], which still enable model heterogeneity as long as there is consensus on attention map shape. FedAD is a one-shot federated learning framework, which means that clients do not have to distill their local model at the beginning of each round, and can participate asynchronously. FedGKT [He *et al.*, 2020] uses intermediate features taking advantage of both asynchronous split learning paradigm [Poirot *et al.*, 2019] and regular FL. Edge devices train small networks composed of a feature extractor, which produces intermediate feature maps, and a classifier, which produces soft targets. Similarly, the server leverages a deeper network and a classifier. After local training, for each local samples, clients communicate their computed intermediate features, the predicted soft targets and the related ground truth labels. The server takes locally computed extracted features as input for its deeper network and produces global soft targets. Both clients and server use a linear combination of regular cross-entropy loss and KD-based loss as objective function. The first considers soft targets and ground truth labels, the latter measures the discrepancy among local and global logits. A similar framework is implemented and extended in FedDKC [Wu *et al.*, 2022b], where Wu *et al.* also develop server-side knowledge refinement strategies.

3.3 Comparison and Adoption Guidelines

Federated adaptations of CD can enable model heterogeneity, and can reduce the communication requirements at the cost of computation overhead with respect to parameter-based schemes. Hence, despite being extremely communication efficient, it may be not always possible to deploy such algorithms on resource-constrained devices due to the overhead of client-side distillation (in Table 1, solutions which use *digestion* at client side), while being a suitable model-

agnostic alternative for cross-silo settings.¹ Furthermore, this class of solutions usually performs worse than FedAvg-based baselines (in terms of global model accuracy) [Sattler *et al.*, 2021a] – even though they typically improve the performance of non-collaborative training [Itahara *et al.*, 2021]. Moreover, most works in this category suppose the existence of a semantically-similar proxy dataset (in some cases even labeled), which may be an unrealistic assumption in some deployment scenarios and use cases (e.g., for specific medical applications). The pioneering communication-efficient data-free strategy in [Jeong *et al.*, 2018] does not incur in local computation overhead, but it is far from achieving global model test accuracy comparable to FedAvg, as demonstrated in [Zhu *et al.*, 2021], also disclosing possible privacy-sensitive information about private data (i.e., per-label model outputs). Solutions as [He *et al.*, 2020; Wu *et al.*, 2022b] enable model heterogeneity, are usually more communication efficient than FedAvg, and include resource-constrained devices in the federation, by adopting a split-learning paradigm and by taking advantage of KD-based regularization. However, as shown in Table 1, due to their split-learning approach, these solutions disclose local ground-truth labels, which again may incur in privacy violation.

4 Tackling FL Data Heterogeneity via KD

KD-based solutions can be used to improve the generalization performance of the global model in presence of data heterogeneity. Server-side approaches rectify FedAvg’s global model either via ensemble distillation on a proxy dataset or using a data-free generator. Orthogonally, client-side mechanisms limit local overfitting or directly control the phenomenon of client drift by distilling global knowledge via on-device regularizers or synthetically-generated data.

4.1 Server-side Refinement of Global Model

Refinement on Publicly Available Data. In [Lin *et al.*, 2020], the authors propose FedDF, a server-side ensemble distillation approach to both enable model heterogeneity and enhance FedAvg’s aggregation. In FedDF, the global model is fine tuned imitating the ensemble (e.g., weighted average) of clients’ model output on a proxy dataset. FedAUX [Sattler *et al.*, 2021a] boosts the performances of FedDF [Lin *et al.*, 2020] leveraging unsupervised pre-training on auxiliary data to find a suitable model initialization for client-side feature extractor. In addition, FedAUX weights the ensemble predictions on the proxy data according to (ϵ, δ) -differentially private [Dwork *et al.*, 2014] certainty score of each participant model. FedBE [Chen and Chao, 2020] proposes to combine client predictions by means of a Bayesian model ensemble to further improve robustness of the aggregation.

Data-free Generator. While server-side ensemble distillation approaches suppose the existence of a proxy dataset, FedFTG [Zhang *et al.*, 2022b] performs a server-side refinement of the global model via data-free knowledge distillation where the server adversarially trains both a generator model and the global model, and fine-tunes the latter

¹For example, Sattler *et al.* use 80000 data points from a public dataset to distill on-device model before local training.

with synthetic data. A data-free generator-based refinement of global model is also proposed in [Zhang *et al.*, 2022a]. The recently proposed DaFKD [Wang *et al.*, 2023] leverages both a server-side and a client-side generator. At each FL round, the server refines the FedAvg-aggregated global model using an importance-weighted ensemble of client models as teacher. The refined global model is distilled with a loss of the form:

$$\mathcal{L} = KL\left(\sum_k^K \alpha_k \cdot \hat{y}^{w^k}, \hat{y}^w\right), \quad (2)$$

with the activated K clients indexed by k , and \hat{y}^{w^k} and \hat{y}^w being, respectively, the prediction of client k 's model and of the global model on synthetically generated data. α_k weights per-sample importance of client models' predictions according to a discriminator model, which is adversarially trained by each client to distinguish whether the synthetic data is sampled from the distribution of its local data.

4.2 Client-side regularization

Local Regularization to Reduce Overfitting. In [Mendieta *et al.*, 2022], Mendieta *et al.* show that GradAug [Yang *et al.*, 2020], a distillation-based structural regularization not specifically designed for FL settings, effectively mitigates client drift issues though substantially introducing computation overhead. Hence, Mendieta *et al.* design a novel method, FedAlign [Mendieta *et al.*, 2022], which has similar effect and performances but with a sustainable computation overhead. In particular, FedAlign targets the deeper layers of a neural network, most prone to overfit client distribution [Luo *et al.*, 2021], imposing a distillation-based term in the local objective function. Such term minimizes the discrepancy among the intermediate features produced in output by the final block of the full network and the features produced by the same block but at reduced width (via temporary uniform pruning). Employing a slimmed sub-block permits to introduce a limited computation overhead.

Local-Global Distillation via Regularization Term. Respectively inspired by fine-tuning optimization ideas and continual learning research, the recent works in [Yao *et al.*, 2023; Ni *et al.*, 2022] and [Lee *et al.*, 2022] find that local KD-based regularization is an effective way of reducing the influence of non-IID data in FL settings. In local-global distillation, the local objective function of clients becomes a linear combination between the cross-entropy loss and a KD-based loss,

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{CE}(\tilde{y}^{w_{t+1}^k}, y) + \lambda\tau^2\mathcal{L}_{KD}(\tilde{y}^{w_{t+1}^k}, \tilde{y}^{w_t}) \quad (3)$$

where $\tilde{y}^{w_{t+1}^k}$ and \tilde{y}^{w_t} are the soft targets produced on local data respectively by the local model of client k and by the received global model. The KD-based loss 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 KL divergence, and works as a regularization term. In this way, the global model works as an anchor for local training, limiting client drift. Local-global distillation can be seen as a self-distillation mechanism, since the local and global model coincide at the beginning of local training. Fig. 3 depicts the basic framework for this kind of local-global distillation.

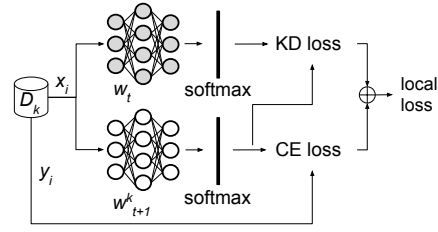


Figure 3: Local-global distillation using a regularization term. w_t represents the global model at round t . w_{t+1}^k is the local model.

Local-Global Distillation via Regularization Term: Further Improvements. FedGKD [Yao *et al.*, 2023] uses an ensemble of M historical global models as teacher, computed as the average of M past global models. Yao *et al.* also propose FedGKD-VOTE as a variation that considers the averaged discrepancy of all the M historical models' outputs as the regularization term. With $M = 1$, the communication cost is the same of FedAvg, while for $M > 1$ the server-client communication cost is doubled, and for FedGKD-VOTE it scales with M . To reduce forgetting among subsequent rounds of learning, FedNTD [Lee *et al.*, 2022] ignores the logits produced by the true classes when computing the softmax score later fed to the KD-based loss. FedLMD [Lu *et al.*, 2023] extends FedNTD by masking out the locally most represented classes in the teacher's logits. Lu *et al.* also proposes a teacher-free version, uniformly distributing the teacher's output probability on minority classes. Similarly, Guo *et al.* design FedED that preserves the global knowledge on not locally represented classes (classes for which a client does not hold samples) [Guo *et al.*, 2024]. He *et al.* further observe that, in the framework of Fig. 3, leveraging an inaccurate global model (i.e., inaccurate teacher) on specific classification classes might mislead local training [He *et al.*, 2022b]. To alleviate such phenomenon, a class-wise adaptive weight is proposed in FedCAD [He *et al.*, 2022b] to control the impact of distillation: when the global model is accurate on a certain class, local models learn more from the distilled knowledge. FedCAD determines the class-wise adaptive weight based on the performances of the global model on an auxiliary dataset, and the server broadcasts such information along with model parameters round by round. FedSSD [He *et al.*, 2022a] extends FedCAD by also considering the credibility of global model at the instance level when computing the distillation term in local training. FedMLB [Kim *et al.*, 2022] enhances the local-global distillation also using intermediate representations, preventing them from deviating too significantly during local fine tuning. To this end, FedMLB crafts hybrid pathways composed of local and global subnetworks, i.e., of local network blocks followed by non-trainable global blocks. Besides regular cross-entropy, local learning also considers the average cross-entropy from hybrid paths and the average KL divergence between the outputs produced by the hybrid paths and the main path as regularization term. FedBR, proposed in [Liu *et al.*, 2023b], extends FedMLB by adaptively distributing only a fraction of the global model's blocks according to the estimated level of data heterogeneity and on the client's computation/communication constraints.

FedDistill⁺, used as alternative baseline in [Yao *et al.*, 2023; Zhu *et al.*, 2021], extends the work of [Jeong *et al.*, 2018; Seo *et al.*, 2020] by exchanging per-label local logits on training dataset in addition to model parameters. With respect to the framework in Fig. 3, FedDistill⁺ uses the received per-label globally averaged logits instead of the output of the global model on private data to calculate the KD loss.

Local-global Distillation via Data-Free Generator Models. Differently from the other work in this subsection, FedGen [Zhu *et al.*, 2021] learns a lightweight server-side generator which is distributed, round by round, to clients that sample it to obtain augmented training examples, using global knowledge as inductive biases in local learning.

4.3 Comparison and Adoption Guidelines

KD-based server-side refinement strategies such as [Lin *et al.*, 2020; Sattler *et al.*, 2021a; Chen and Chao, 2020] can improve FedAvg global model performance in presence of highly heterogeneous data when semantically-similar unlabeled proxy data are available. It is worth noting that this class of algorithms exhibits most improvements when several local epochs are performed between communication rounds and client models tend to drift apart. Also, such algorithms do not introduce computation or communication overhead on clients. Data-free generator models can also be used to perform server-side global model corrections as in [Zhang *et al.*, 2022b] or to limit client drift directly at the participating devices as in [Zhu *et al.*, 2021], in both cases at the cost of disclosing local label count. Similarly, DaFKD [Wang *et al.*, 2023] enables data-free refinement of global model via generators, but introducing computation and communication overheads on clients – although limited thanks to partial parameter sharing.

No additional information has to be disclosed from clients and not even proxy data are needed in solutions that regularize local training by employing global model output on local data, as in FedGKD [Yao *et al.*, 2023], FedNTD [Lee *et al.*, 2022], FedLMD [Lu *et al.*, 2023], FedED [Guo *et al.*, 2024]. In addition, this set of strategies do not introduce significant on-device computation overhead and has the same communication requirements as FedAvg. However, they would require to store two full-size models in memory (the local model as usual and the global model as reference), such limitation can, in practice, be avoided by firstly computing the predictions of the received global model on local data, and then proceeding with local training by overwriting it. If limited labeled proxy data are available, local-global knowledge distillation can be improved as in [He *et al.*, 2022b; He *et al.*, 2022a]. When moderate or even significant computing overhead is sustainable, local global distillation can be enhanced by using intermediate features and hybrid pathways as in FedMLB [Kim *et al.*, 2022] and FedBR [Liu *et al.*, 2023b], thus significantly improving the effectiveness of local-global distillation. It is worth noting that in FedMLB and FedBR the selected blocks of the global model must be stored locally during training, and this cannot be avoided as for regularization based on model responses. As highlighted in [Kim *et al.*, 2022], client-side regularization can be coupled with

standard server-side strategies to boost performances (e.g., FedOpt [Reddi *et al.*, 2020]).

5 Emerging Use of KD in FL

Here we outline the recently emerging use of KD tailored to address other FL issues, i.e., personalized FL, federated unlearning, and class-incremental federated learning.

KD for FL Personalization. The primary scope of personalized FL is to build a global model prone to quickly adapt to local data distributions. While KD-based solutions are commonly used to enhance the generalization ability of the global model (see Section 4), lately, distillation-inspired mechanisms have been employed in personalized FL algorithms. Jin *et al.* propose pFedSD where clients store their last trained personalized model and use it as teacher for the next rounds [Jin *et al.*, 2022]. In a nutshell, pFedSD uses the distillation framework depicted in Fig. 3, with the teacher being a past local model instead of the global model. In addition, Chen *et al.* recently introduced a *spectral* distillation used to regularize local training via a term that minimizes the KL divergence between the Fourier spectra of global and personalized models [Chen *et al.*, 2023].

KD for Federated Unlearning. An FL client should have the right to request the removal of its contribution from the global model, and mechanisms to enable selective forgetting of clients are emerging. The solution in [Wu *et al.*, 2022a] firstly removes the unlearning client’s historical model updates from the global model. Then, the server leverages KD to quickly recover the global model performance with the sanitized global model mimicking the outputs of the last global model on proxy unlabeled data.

KD for Federated Class-Incremental Learning. The methods surveyed up to here assume that the classification classes of a task are fixed over the entire FL process. In contrast, works to address class-incremental learning in federated settings are emerging. In Liu *et al.*, [Liu *et al.*, 2023a] propose FedET, which leverages an enhancer distillation method to modify the imbalance between old and new knowledge. In [Wu *et al.*, 2024], Wu *et al.* design FedNASD, a method that uses the class probabilities from the current models to approximate the historical models’ output on new classes – historical models may encompass fewer classes – so that local-global distillation (see Fig. 3) can be applied even when classes are changing over time.

6 Conclusive Remarks

KD has been recently embedded in FL algorithms to tackle FL-specific issues. In this survey, we present the various possible distillation-based mechanisms primarily proposed to enable model heterogeneity and to tackle the effects of data heterogeneity in FL, and, in some cases, to reduce the associated communication costs. We originally classify and compare the recent proposals in this field, by highlighting their possible weaknesses and their trade-offs. Then, we outline the emerging use of KD to improve other aspects of FL.

References

- [Anil *et al.*, 2018] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018.
- [Bellavista *et al.*, 2021] Paolo Bellavista, Luca Foschini, and Alessio Mora. Decentralised learning in federated deployment environments: A system-level survey. *ACM Computing Surveys (CSUR)*, 54(1):1–38, 2021.
- [Chang *et al.*, 2019] Hongyan Chang, Virat Shejwalkar, Reza Shokri, and Amir Houmansadr. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. *arXiv preprint arXiv:1912.11279*, 2019.
- [Chen and Chao, 2020] Hong-You Chen and Wei-Lun Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. *arXiv preprint arXiv:2009.01974*, 2020.
- [Chen *et al.*, 2023] Zihan Chen, Howard Hao Yang, Tony Quek, and K. F. Ernest Chong. Spectral co-distillation for personalized federated learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [Cheng *et al.*, 2021] Sijie Cheng, Jingwen Wu, Yanghua Xiao, and Yang Liu. Fedgems: Federated learning of larger server models via selective knowledge fusion. *arXiv preprint arXiv:2110.11027*, 2021.
- [Diakonikolas *et al.*, 2017] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. In *International Conference on Machine Learning*, pages 999–1008. PMLR, 2017.
- [Dwork *et al.*, 2014] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.
- [Gong *et al.*, 2021] Xuan Gong, Abhishek Sharma, Srikrishna Karanam, Ziyang Wu, Terrence Chen, David Doermann, and Arun Innanje. Ensemble attention distillation for privacy-preserving federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15076–15086, 2021.
- [Gou *et al.*, 2021] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.
- [Guo *et al.*, 2024] Kuangpu Guo, Yuhe Ding, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Not all minorities are equal: Empty-class-aware distillation for heterogeneous federated learning. *arXiv preprint arXiv:2401.02329*, 2024.
- [He *et al.*, 2020] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in Neural Information Processing Systems*, 33:14068–14080, 2020.
- [He *et al.*, 2022a] Yuting He, Yiqiang Chen, Xiaodong Yang, Hanchao Yu, Yi-Hua Huang, and Yang Gu. Learning critically: Selective self-distillation in federated learning on non-iid data. *IEEE Transactions on Big Data*, 2022.
- [He *et al.*, 2022b] Yuting He, Yiqiang Chen, Xiaodong Yang, Yingwei Zhang, and Bixiao Zeng. Class-wise adaptive self distillation for heterogeneous federated learning. 2022.
- [Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [Hu *et al.*, 2021] Li Hu, Hongyang Yan, Lang Li, Zijie Pan, Xiaozhang Liu, and Zulong Zhang. Mhat: an efficient model-heterogeneous aggregation training scheme for federated learning. *Information Sciences*, 560:493–503, 2021.
- [Itahara *et al.*, 2021] Sohei Itahara, Takayuki Nishio, Yusuke Koda, Masahiro Morikura, and Koji Yamamoto. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Transactions on Mobile Computing*, 22(1):191–205, 2021.
- [Jeong *et al.*, 2018] Eunjeong Jeong, Seungeun Oh, Hye-sung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.
- [Jin *et al.*, 2022] Hai Jin, Dongshan Bai, Dezhong Yao, Yutong Dai, Lin Gu, Chen Yu, and Lichao Sun. Personalized edge intelligence via federated self-knowledge distillation. *IEEE Transactions on Parallel and Distributed Systems*, 34(2):567–580, 2022.
- [Karimireddy *et al.*, 2020] Sai P Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and A Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.
- [Kim *et al.*, 2022] Jinkyu Kim, Geeho Kim, and Bohyung Han. Multi-level branched regularization for federated learning. In *International Conference on Machine Learning*, pages 11058–11073. PMLR, 2022.
- [Lee *et al.*, 2022] Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. Preservation of the global knowledge by not-true distillation in federated learning. In *Advances in Neural Information Processing Systems*, 2022.
- [Li and Wang, 2019] Daliang Li and Junpu Wang. Fedmd: Heterogeneous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.
- [Lin *et al.*, 2020] 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.
- [Liu *et al.*, 2023a] Chenghao Liu, Xiaoyang Qu, Jianzong Wang, and Jing Xiao. Fedet: a communication-efficient

- federated class-incremental learning framework based on enhanced transformer. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 3984–3992, 2023.
- [Liu *et al.*, 2023b] Jianchun Liu, Qingmin Zeng, Hongli Xu, Yang Xu, Zhiyuan Wang, and He Huang. Adaptive block-wise regularization and knowledge distillation for enhancing federated learning. *IEEE/ACM Transactions on Networking*, 2023.
- [Lu *et al.*, 2023] Jianghu Lu, Shikun Li, Kexin Bao, Pengju Wang, Zhenxing Qian, and Shiming Ge. Federated learning with label-masking distillation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 222–232, 2023.
- [Luo *et al.*, 2021] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34:5972–5984, 2021.
- [McMahan *et al.*, 2016] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.
- [Mendieta *et al.*, 2022] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8397–8406, 2022.
- [Ni *et al.*, 2022] Xuanming Ni, Xinyuan Shen, and Huimin Zhao. Federated optimization via knowledge codistillation. *Expert Systems with Applications*, 191:116310, 2022.
- [Papernot *et al.*, 2016] Nicolas Papernot, Martín Abadi, Ul- far Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.
- [Poirot *et al.*, 2019] Maarten G Poirot, Praneeth Vepakomma, Ken Chang, Jayashree Kalpathy-Cramer, Rajiv Gupta, and Ramesh Raskar. Split learning for collaborative deep learning in healthcare. *arXiv preprint arXiv:1912.12115*, 2019.
- [Reddi *et al.*, 2020] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.
- [Sattler *et al.*, 2019] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 2019.
- [Sattler *et al.*, 2021a] 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.
- [Sattler *et al.*, 2021b] Felix Sattler, Arturo Marban, Roman Rischke, and Wojciech Samek. Cfd: Communication-efficient federated distillation via soft-label quantization and delta coding. *IEEE Transactions on Network Science and Engineering*, 2021.
- [Selvaraju *et al.*, 2017] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.
- [Seo *et al.*, 2020] Hyowoon Seo, Jihong Park, Seungeun Oh, Mehdi Bennis, and Seong-Lyun Kim. Federated knowledge distillation. *arXiv preprint arXiv:2011.02367*, 2020.
- [Wang *et al.*, 2023] Haozhao Wang, Yichen Li, Wenchao Xu, Ruixuan Li, Yufeng Zhan, and Zhigang Zeng. Dafkd: Domain-aware federated knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20412–20421, 2023.
- [Wu *et al.*, 2022a] Chen Wu, Sencun Zhu, and Prasenjit Mitra. Federated unlearning with knowledge distillation. *arXiv preprint arXiv:2201.09441*, 2022.
- [Wu *et al.*, 2022b] Zhiyuan Wu, Sheng Sun, Yuwei Wang, Min Liu, Quyang Pan, Junbo Zhang, Zeju Li, and Qingxiang Liu. Exploring the distributed knowledge congruence in proxy-data-free federated distillation. *ACM Transactions on Intelligent Systems and Technology*, 2022.
- [Wu *et al.*, 2024] Zhiyuan Wu, Tianliu He, Sheng Sun, Yuwei Wang, Min Liu, Bo Gao, and Xuefeng Jiang. Federated class-incremental learning with new-class augmented self-distillation. *arXiv preprint arXiv:2401.00622*, 2024.
- [Yang *et al.*, 2020] Taojiannan Yang, Sijie Zhu, and Chen Chen. Gradaug: A new regularization method for deep neural networks. *Advances in Neural Information Processing Systems*, 33:14207–14218, 2020.
- [Yao *et al.*, 2023] Dezhong Yao, Wanning Pan, Yutong Dai, Yao Wan, Xiaofeng Ding, Chen Yu, Hai Jin, Zheng Xu, and Lichao Sun. Fedgkd: Towards heterogeneous federated learning via global knowledge distillation. *IEEE Transactions on Computers*, 2023.
- [Zhang *et al.*, 2022a] Lan Zhang, Dapeng Wu, and Xiaoyong Yuan. Fedzkt: Zero-shot knowledge transfer towards resource-constrained federated learning with heterogeneous on-device models. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, pages 928–938. IEEE, 2022.
- [Zhang *et al.*, 2022b] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10174–10183, 2022.
- [Zhu *et al.*, 2021] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning*, pages 12878–12889. PMLR, 2021.