One-shot Federated Learning Methods: A Practical Guide

Xiang Liu[1] , **Zhenheng Tang**[2] , **Xia Li**[3] , **Yijun Song**[4] , **Sijie Ji**[5] ,
Zemin Liu[6] , **Bo Han**[7,†] , **Linshan Jiang**[1] and **Jialin Li**[1]

[1]National University of Singapore, [2]The Hong Kong University of Science and Technology,
[3]Department of Information Technology and Electrical Engineering, ETH Zürich,
[4]Zhejiang University of Finance & Economics Dongfang College, [5]California Institute of Technology
[6]College of Computer Science and Technology, Zhejiang University,
[7]Department of Computer Science, Hong Kong Baptist University
† Corresponding Authors

{liuxiang,lijl}@comp.nus.edu.sg, zhtang.ml@ust.hk, xiaxiali@ethz.ch, yijunsong@zufedfc.edu.cn,
sijieji@caltech.edu, liu.zemin@zju.edu.cn, bhanml@comp.hkbu.edu.hk, linshan@nus.edu.sg

Abstract

One-shot Federated Learning (OFL) is a distributed machine learning paradigm that constrains client-server communication to a single round, addressing privacy and communication overhead issues associated with multiple rounds of data exchange in traditional Federated Learning (FL). OFL demonstrates the practical potential for integration with future approaches that require collaborative training models, such as large language models (LLMs). However, current OFL methods face two major challenges: data heterogeneity and model heterogeneity, which result in subpar performance compared to conventional FL methods. Worse still, despite numerous studies addressing these limitations, a comprehensive summary is still lacking. To address these gaps, this paper presents a systematic analysis of the challenges faced by OFL and thoroughly reviews the current methods. We also offer an innovative categorization method and analyze the trade-offs of various techniques. Additionally, we discuss the most promising future directions and the technologies that should be integrated into the OFL field. This work aims to provide guidance and insights for future research.

1 Introduction

Federated Learning (FL) [McMahan *et al.*, 2016] is an established paradigm that fosters multiple clients to collaboratively engage in distributed machine learning, allowing the aggregation of local models on a server to generate a global model. In practice, current implementations of FL require multiple rounds of data exchange between clients and the server to ensure the training of a high-accuracy global model without the need to directly utilize raw data. However, as FL is widely employed in fields such as intelligent transportation, economy, manufacturing and healthcare [Kairouz *et al.*, 2021], the traditional

multi-round setup can still violate privacy-preserving principles [Kairouz *et al.*, 2021] and incur significant communication overhead [Wang *et al.*, 2023; Tang *et al.*, 2024b].

To protect data privacy, researchers have proposed three main techniques to address privacy leakage in FL: Differential Privacy (DP) [Dwork, 2011], Secure Multi-Party Computation (SMPC) [Bonawitz *et al.*, 2017], and Homomorphic Encryption (HE) [Cheon *et al.*, 2017]. DP-based methods introduce noise into the intermediate data before sharing to prevent privacy leakage, but this can reduce model accuracy. SMPC-based methods increase network communication overhead between participants, especially when scaling to a large number of clients. HE-based models are criticized for requiring substantial computational resources. While these methods address privacy-preserving issues, they do not tackle communication overhead and can introduce additional issues.

Consequently, communication between server and clients is limited to a single round, as in one-shot Federated Learning (OFL) [Guha *et al.*, 2019]. OFL methods have been specifically designed to improve accuracy in one-shot scenarios. OFL not only reduces the burden of communication transmission but also potentially achieves even stronger security due to its single-round setting when integrated with existing privacy-preserving methods. Additionally, it offers further advantages, such as alleviating the requirements for transmission synchronization [Alemdar *et al.*, 2021]. Therefore, OFL methods have emerged as a promising approach to address these practical issues. Since 2019, a significant number of researchers, recognizing the potential of OFL, have begun to study this area and have already proposed various methods aimed at addressing specific challenges as well as enhancing global model performance on the server side [Tang *et al.*, 2024b; Shen *et al.*, 2025].

However, none of the current papers has effectively summarized the challenges within the OFL domain, and most researchers have provided inadequate summaries and classifications of OFL-related literature. This may be due to the rapid development of OFL as a practical field, which has yet to receive a thorough overview. Additionally, because current OFL methods often employ multiple techniques simultane-

ously, categorization becomes muddled, leading to inconsistent and imprecise classification approaches in the literature.

Existing survey papers tend to focus on Knowledge Distillation [Mora *et al.*, 2024], Label Leakage [Liu *et al.*, 2024b], or integration with Foundation Models [Woisetschlager and others, 2024] in FL, but none have concentrated on providing a comprehensive overview of the challenging and promising practical methods in OFL. As Large Language Models (LLMs) gain prominence, maintaining data privacy during large-scale model training and reducing communication overhead through collaborative training among multiple clients becomes increasingly important [Tang *et al.*, 2024a; Ye *et al.*, 2024]. Especially with the development of cloud-edge collaborative frameworks [Tang *et al.*, 2024a], OFL may demonstrate even broader applicability alongside LLMs.

Thus, we present the first survey paper specifically focused on the OFL domain. Our main contributions are not limited to a standard survey; they are summarized as follows:

- We propose novel taxonomies addressing the challenges faced by OFL and current methods, aiding researchers in better understanding the issues and existing approaches.

- We provide a thorough and detailed overview of existing OFL methods, including their primary designs, underlying principles, and advantages and potential drawbacks in addressing specific challenges. This enables researchers to gain a deeper understanding of approaches.

- Beyond summarizing existing methods, we discuss our findings, highlight promising future directions, and identify relevant areas, assisting researchers in integrating OFL with practical applications.

The paper is organized as follows. Section 2 introduces the fundamental challenges of one-shot Federated Learning. Section 3 provides a novel taxonomy and discusses in detail the technical aspects of each category within the context of OFL. Section 4 offers a discussion that summarizes our findings and outlines future directions for the development of OFL methods. Finally, Section 5 concludes our paper.

2 Challenges in One-shot Federated Learning

2.1 Problem Formulation

In federated learning, clients collaboratively train local models using their private local datasets to produce a global model for the server within a distributed fashion. The objective is to minimize the optimization problem:

$$\min_{\boldsymbol{w}} F(\boldsymbol{w}) := \alpha_i \sum_{i \in [n]} F_i(\boldsymbol{w_i}) \tag{1}$$

where n denotes the number of clients, $\boldsymbol{w}_i$ denotes the parameters of local client$_i$, $\boldsymbol{w}$ denotes the parameters of the global model, α_i denotes the proportion of client$_i$'s private dataset to the entire dataset, F is the loss function.

In the traditional FL setting, clients upload their parameters $\boldsymbol{w}_i$ multiple times to get the final $\boldsymbol{w}$. However, in OFL, they only upload their parameters once.

2.2 Proposed Taxonomy for Challenges

Besides the privacy issues, OFL methods must improve the global model's test accuracy while addressing two main challenges: (1) **Data Heterogeneity**, where the private datasets on each client are non-iid (independent and identically distributed); and (2) **Model Heterogeneity**, as the local models participating in FL often differ from one another.

Data Heterogeneity

In practice, data heterogeneity arises from the non-IID characteristics of the private datasets among clients, which are primarily due to three types of skew [Li *et al.*, 2022]: quantity skew, feature skew and label skew.

For quantity skew, each client has the same data distribution but with different amounts of data. For feature skew, the classes within different clients' private datasets have different statistics. For label skew, different clients have varying proportions of data for the same class.

In traditional FL, numerous methods have been proposed to handle data heterogeneity, such as FedAvg [McMahan *et al.*, 2016] (averaging parameters from clients), SCAFFOLD [Karimireddy *et al.*, 2020] and MOON [Li *et al.*, 2021a] (measuring the similarity and disparity of parameters between parties), FedProx [Li *et al.*, 2020] (adding an L2 regularization term for optimization) and VHL [Tang *et al.*, 2022] (calibrating features via virtual data). These methods work well in non-one-shot scenarios by iteratively updating and integrating the features of client models with the server round by round [Li *et al.*, 2022].

However, in a one-shot setting with non-IID data, the differences in model data across clients can be substantial, posing new challenges that traditional methods struggle to address efficiently in a single pass. This challenge is essentially an out-of-distribution (OOD) detection [Yang *et al.*, 2024a] problem. Since models are uploaded only once, if the FL method fails to capture all the hidden dataset statistics in the client parameters effectively, parts of the data samples may be OOD in the global model and remain unrecognized [Guha *et al.*, 2019]. In multi-round settings, global models can be finetuned through iterative updates, but OFL methods require more robust learning capabilities to overcome the data heterogeneity challenge without sacrificing accuracy.

Model Heterogeneity

In practice, having all clients and the server share the same model as the global model often fails to meet the diverse needs of all clients, especially since FL is commonly integrated with different model architectures within cloud-edge collaborative frameworks or deployed on different edge devices, or due to privacy considerations [Zhang *et al.*, 2024a]. Consequently, personalized FL [Smith *et al.*, 2017] was proposed to address these concerns, and most of this research is based on the assumption of model homogeneity.

Model heterogeneity arises from differences in the communication capabilities [Shah and Lau, 2021], computing resources [Shin *et al.*, 2024], and model architectures of local clients [Shen *et al.*, 2025]. Even when clients share the

same model, disparities in computing resources can lead to different clients using different sub-models from the global model [Diao *et al.*, 2021; Shen *et al.*, 2025] for training or updating only specific layers' parameters to meet varying requirements [Li *et al.*, 2021c], resulting in model heterogeneity. Additionally, clients may be unwilling to disclose details of their model designs, further complicating the issue.

Therefore, in OFL tasks, researchers must consider not only data heterogeneity but also the persistent challenge of model heterogeneity. Current approaches for model heterogeneity of OFL primarily concentrate on addressing the challenge posed by differing model architectures among clients.
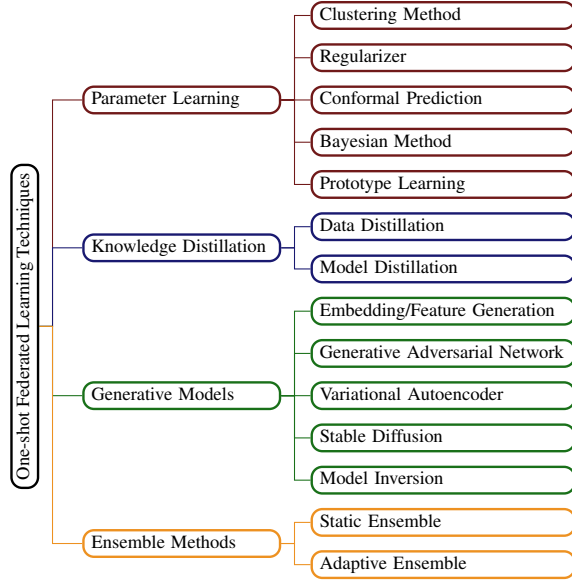

Figure 1: A basic taxonomy of one-shot federated learning techniques. Note that Some hybrid methods employ multiple techniques.

3 One-shot Federated Learning Techniques

Figure 1 illustrates our new taxonomy of current OFL techniques. These techniques aim to enhance the final global model's accuracy in light of the two challenges we previously mentioned. From an innovative perspective, we comprehensively categorize all the current methods into four main groups based on the techniques they employ: **Parameter Learning, Knowledge Distillation, Generative Models, and Ensemble Methods**. Besides these four techniques, many methods are hybrid methods, which employ a combination of multiple techniques.

3.1 Parameter Learning

Methods in the parameter learning category are derived from approaches like FedAvg and tend to directly learn the statistical information of local clients' model parameters to determine the global model's parameters.

Clustering Method. k-FED [Dennis *et al.*, 2021] employs Lloyd's k-means clustering method to first cluster the information from local models and then uploads the cluster means to the server. This aids the server in learn-

ing more local information in a one-shot scenario, albeit with increased communication costs. **Regularizer.** MA-Echo [Su *et al.*, 2023] considers aggregation of local models' parameters layer-wise and uses a unique method by introducing new norms. This approach helps the global model better account for differences among local models. **Conformal Prediction.** FedCP-QQ [Humbert *et al.*, 2023] utilizes a conformal prediction method, demonstrating that for any distribution, a prediction set with the desired coverage can be computed in a single round of communication, thereby improving the global model's performance. **Bayesian Method.** FedFisher [Jhunjhunwala *et al.*, 2023] is an aggregation method that uses the empirical Fisher information matrix obtained by layer-wise Laplace approximation. This method first leverages Bayesian techniques to better model the posteriors of local clients and then aggregates for the global model. FedLPA [Liu *et al.*, 2024a] employs a similar approach to FedFisher, further directly training the global model's parameters to achieve results. β-PredBaye [Hasan *et al.*, 2024] first collects parameters from the posteriors of local clients and then introduces a tunable hyper-parameter β. This parameter interpolates between a mixture and a product of the predictive posteriors by considering merging Gaussians in the predictive space, and aggregation is performed based on this approach. **Prototype Learning.** In prototype learning [Snell *et al.*, 2017], one or more prototypes for each class are calculated using the samples from the training set. These prototypes can be considered as central representations summarizing and representing each class. PNFM [Yurochkin *et al.*, 2019] employs Bayesian Nonparametric Methods for neural matching based on prototype learning. FedMA [Wang *et al.*, 2020] extends the approach from PNFM by moving beyond fully-connected networks. It constructs a shared global model in a hierarchical manner by matching and averaging hidden elements (such as channels in convolutional layers or hidden states in LSTMs) with similar feature extraction characteristics, thus expanding the methodology to CNNs and LSTMs.

Since methods with parameter learning techniques are based on parameter optimization, they typically provide theoretical proofs of convergence. As these methods require uploading local model parameters, they can introduce privacy concerns. As a result, many approaches include ablation studies that integrate with Differential Privacy (DP). This category of methods can address the data heterogeneity challenge, potentially improving the global model's accuracy over traditional FL methods. However, due to the inherent limitations in accurately modeling the statistics, there is still a need to explore better approaches for capturing the features of local clients' parameters. Additionally, parameter learning techniques do not address the model heterogeneity challenge. Note that within parameter learning, prototype learning holds the most promise for integration with others.

3.2 Knowledge Distillation

Knowledge distillation is a compression technique aimed at reducing the model's size and computational demands while maintaining as much accuracy as possible. It has been shown to address OFL problems effectively and is categorized into

Method	Distillation	Generative Models	Ensemble	Additional Data
DOSFL [Zhou *et al.*, 2020]	Data	-	-	-
FedD3 [Song *et al.*, 2023]	Data	-	-	-
Fusion Learning [Kasturi *et al.*, 2020]	-	Data Distribution Digest	-	-
XorMixFL [Shin *et al.*, 2020]	-	Embedding	-	Data Labels
FedOCD [Liu *et al.*, 2024c]	-	Embedding	-	-
FedPFT [Beitollahi *et al.*, 2024]	-	Feature Extractor (GMM)	-	Three Foundation Models
OSGAN [Kasturi, 2023]	-	Generative Adversarial Network	-	-
FedDEO [Yang *et al.*, 2024c]	-	Stable Diffusion	-	-
FedBiP [Chen *et al.*, 2024]	-	Stable Diffusion	-	Latent Diffusion Model [Rombach *et al.*, 2022]
FedDISC [Yang *et al.*, 2024b]	-	Stable Diffusion	-	CLIP [Radford *et al.*, 2021]
FedMITR [Hao *et al.*, 2025]	-	Model Inversion	-	-
DeDES [Wang *et al.*, 2023]	-	-	Static	-
FedTMOS [Qi *et al.*, 2025]	-	-	Static	-
FENS [Allouah *et al.*, 2024]	-	-	Adaptive	-
FuseFL [Tang *et al.*, 2024b]	-	-	Adaptive	-
HPFL [Shen *et al.*, 2025]	-	-	Adaptive	-
Mediator [Lai *et al.*, 2025]	-	-	Adaptive	-
FedOV [Diao *et al.*, 2023]	-	Generative Adversarial Network	Static	-
IntactOFL [Zeng *et al.*, 2024]	-	Generative Adversarial Network	Adaptive	-
FedSD2C [Zhang *et al.*, 2024b]	Data	Variational Autoencoder	-	-
One-shot FL [Guha *et al.*, 2019]	Model $\rightarrow$	-	Static	Public Dataset
FedKT [Li *et al.*, 2021b]	Model $\rightarrow$	-	Static	Public Dataset
FedKD [Gong *et al.*, 2022]	Model $\leftarrow$	-	Static	Public Dataset
Dense [Zhang *et al.*, 2022]	Model $\leftarrow$	Generative Adversarial Network	Static	-
Co-Boosting [Dai *et al.*, 2024]	Model $\leftarrow$	Generative Adversarial Network	Adaptive	-
FedCVAE [Heinbaugh *et al.*, 2023]	Model	Variational Autoencoder	Static	Data Labels

Table 1: A synoptic overview of the surveyed solutions includes **Knowledge Distillation**, **Generative Models**, **Ensemble Methods**, and **Hybrid Methods**. The dashed line distinguishes between different categories. "Additional data" refers to the extra data requirements needed. In the distillation column, "$\rightarrow$" indicates that distillation is performed before applying the ensemble method, whereas "$\leftarrow$" suggests the opposite, with distillation based on the ensemble model.

two major types: data distillation and model distillation.

Data Distillation. DOSFL [Zhou *et al.*, 2020] involves distilling the local private dataset at each client and then training on the distilled data at the server. FedD3 [Song *et al.*, 2023] operates in resource-constrained edge environments by collecting decentralized dataset distillation to train the global model. Additionally, FedSD2C [Zhang *et al.*, 2024b] incorporates data distillation alongside other techniques, which we will analyze in Section 3.5. **Model Distillation.** The methods utilize model distillation, and as shown in Table 1, these often incorporate multiple techniques, typically in combination with ensemble methods. Generally, there are two approaches: (1) conducting model distillation first and then ensembling [Guha *et al.*, 2019; Li *et al.*, 2021b], (2) or vice versa—ensembling first and distilling the ensembled model afterward [Gong *et al.*, 2022; Zhang *et al.*, 2022; Dai *et al.*, 2024]. We will discuss these methods in detail in Section 3.5.

Compared to parameter learning techniques, knowledge distillation provides better privacy preservation. By allocating more computational resources to distillation, it iteratively enhances the global model's performance, yielding higher accuracy and supporting model heterogeneity.

3.3 Generative Models

Generative models are a widely used class of techniques in machine learning, designed to train synthetic samples that closely resemble the original data, thereby ensuring similar data distributions. Consequently, generative models can be utilized in OFL to aid the server in training on the synthetic datasets, which is an effective learning from local datasets. **Embedding/Feature Generation.** In Fusion Learning [Kasturi *et al.*, 2020], the distribution digest of client data and their local model parameters are sent to the server. The server regenerates dataset based on these distribution digest, integrates parameters from multiple devices to construct a global model, trains the model using the combined dataset. XorMixFL [Shin *et al.*, 2020] assumes both the global server and clients possess labeled samples from a global class. The server collects encoded data samples (embeddings) from other devices to construct the global data samples. It employs the exclusive OR operation (XOR) to protect privacy during the encoding process. FedOCD [Liu *et al.*, 2024c] focuses on cross-domain recommendation and applies local differential privacy for security while processing the generated user embeddings. FedPFT [Beitollahi *et al.*, 2024] utilizes a feature extractor to derive the corresponding Gaussian mixture models (GMMs) to model synthetic sample features, transferring per-client parametric models. **Generative Adversarial Networks.** OSGAN [Kasturi, 2023] further employs Generative Adversarial Networks (GANs) as a generative model to enhance performance. Other uses [Zhang *et al.*, 2022; Diao *et al.*, 2023; Zeng *et al.*, 2024; Dai *et al.*, 2024] of Generative Adversarial Networks, along with additional techniques, can be found in Table 1 and will be discussed in detail in Section 3.5. **Variational Autoencoder.** Researchers have further utilized Variational Autoencoders in approaches such as FedCAVE [Heinbaugh *et al.*, 2023] and FedSD2C [Zhang *et al.*, 2024b]. **Stable Diffusion.** Fed-

DEO [Yang *et al.*, 2024c] initially adopts popular diffusion models, transmitting image captions (local descriptions) to better capture the distribution of client local data. This assists the server in generating synthetic data that align more closely with the client's data distribution. FedBiP [Chen *et al.*, 2024] further utilizes a better Stable Diffusion model, specifically the foundation model (Latent Diffusion Model [Rombach *et al.*, 2022]), to improve performance. FedDISC [Yang *et al.*, 2024b] leverages a pre-trained model CLIP [Radford *et al.*, 2021] with prototype learning. It involves four steps: Prototype Extraction, Pseudo Labeling, Feature Processing, and Image Generation, which lead to improved outcomes. **Model Inversion.** FedMITR [Hao *et al.*, 2025] employs model inversion techniques to obtain statistical information about some private data within local models. Additionally, it uses a vision transformer to perform token relabeling.

Based on the above discussion, we find that compared to parameter learning, knowledge distillation and generative models offer advantages beyond higher accuracy when given more computational resources. We observe that both techniques address data and model heterogeneity challenges. Additionally, they enhance security by not directly sharing client parameters. Since both techniques aim to statistically approximate the true local datasets without exposing actual data, as local models also reflect local datasets. Thus, they enhance the global model's capability with similar approaches to learning client-specific features, which enables the global model to train more effectively while mitigating out-of-distribution (OOD) issues.

These two techniques also have their differences. Model distillation improves methods by distilling parameter information from clients rather than using their raw data. In contrast, generative models enhance learning by generating synthetic data that capture the characteristics of clients' private datasets. Although data distillation also emphasizes the data aspect, generative models can address the inefficiencies associated with knowledge distillation methods, often leading to superior performance. As a result, numerous researchers are actively pursuing advancements in methods that better reflect local data distributions, especially through generative models. This progression is evident as the focus has shifted from GANs to VAEs, then to Stable Diffusion, and finally to increasingly sophisticated diffusion models.

3.4 Ensemble Methods

Ensemble methods represent one of the most straightforward solutions to addressing OFL challenges (data heterogeneity and model heterogeneity). The most naïve ensemble approach involves aggregating global models obtained from local clients. However, ensemble models can actually be combined with a wide range of other techniques. Currently, these combinations are mainly categorized into two types: static and adaptive ensemble methods.

Static. Static ensemble methods are typically used to perform operations like averaging (DeDES [Wang *et al.*, 2023]) or taking the maximum based on the outputs from local clients. FedTMOS [Qi *et al.*, 2025] enhances this approach by integrating a lightweight reinforcement learning

model, Tsetlin Machine, with prototype learning. Each client can train a different Tsetlin Machine for each class. Due to the unique nature of the Tsetlin Machine, which employs Boolean feature representation, it is well-suited for handling sparse or high-dimensional data. This leads to improved results on the server side. **Adaptive.** FENS [Allouah *et al.*, 2024] balances communication rounds and accuracy by adjusting the ensemble weights based on the similarity between global and local models. From the causality perspective, FuseFL [Tang *et al.*, 2024b] identifies that the performance drop comes from the isolated training problem. Then, FuseFL proposes a bottom-up approach that trains and merges sub-models adaptively to enhance invariant feature learning, thus alleviating spurious fitting. HPFL [Shen *et al.*, 2025] proposes regarding the personalized modules as hot pluggable plug-ins which can be selected according to the input data during inference. The plug-ins only need to be sent to server once. Mediator [Lai *et al.*, 2025] aggregates different finetuned LLMs that can be trained by different parties with low memory occupation and adaptive routing and compression.

Other methods [Guha *et al.*, 2019; Li *et al.*, 2021b; Gong *et al.*, 2022; Zhang *et al.*, 2022; Heinbaugh *et al.*, 2023; Diao *et al.*, 2023; Zeng *et al.*, 2024; Dai *et al.*, 2024] that utilize ensemble techniques are listed in Table 1 and will be discussed in Section 3.5. While ensemble methods directly leverage local models, they can be combined with other techniques to ensure privacy and improve accuracy. We also find that adaptive ensemble methods typically yield better results.

3.5 Hybrid Methods

As previously mentioned, current papers do not adequately classify methods addressing one-shot Federated Learning. Table 1 illustrates the challenges of classification and highlights the effectiveness of our unique classification approach. This paper thoroughly examines all current methods of OFL from a technical contribution perspective. A single method can employ multiple techniques. The three techniques that can tackle model heterogeneity—knowledge distillation, generative models, and ensemble methods, are often integrated to enhance performance and privacy. These methods can be combined in pairs or even all three together. Based on this, and following our detailed discussions of each technique, we summarize these methods, providing an in-depth analysis of how their integration enhances performance.

Integrating Generative Models with Ensemble Methods. FedOV [Diao *et al.*, 2023] addresses the open-set problem with an OOD perspective, where some classes may be missing on certain clients due to label skew. To handle this, it uses GAN to generate outliers with feature corruption and employs adversarial training techniques. The generated data samples serve as additional unknown classes during local training, and the results are obtained by ensemble methods from local clients. IntactOFL [Zeng *et al.*, 2024] also employs GANs to generate synthetic data samples but adopts a more flexible approach by utilizing a mixture of experts (MoE) model [Jacobs *et al.*, 1991]. This allows for dynamic adjustment of the ensemble weights for each local client, tai-

lored specifically to each data instance, which greatly improves the performance.

Integrating Generative Models with Distillation. FedSD2C [Zhang *et al.*, 2024b] begins by using a pretrained Autoencoder to extract distillate synthesis. To reduce communication costs and enhance the privacy of the distillates, clients perturb the distillates using the Fourier transform. To ensure that the distillates fully encompass local information, clients perform a V-information based Core-Set selection when utilizing the encoder. Finally, the server collects all the distillates and uses the pretrained autoencoder's decoder to train an accurate global model.

Integrating Ensemble Methods with Distillation. One-shot FL [Guha *et al.*, 2019], particularly in the context of semi-supervised learning, leverages distillation to reduce the size of the global model and enhance privacy guarantees. The approach then applies average ensemble methods to distilled local models to achieve favorable results. FedKT [Li *et al.*, 2021b], while using a similar approach, extends previous methods from support vector machines (SVMs). It incorporates a hierarchical knowledge transfer framework with a two-tier privacy aggregation of teacher ensembles (PATEs) structure to improve accuracy and introduces the concept of consistent voting to strengthen the ensemble. Both of these methods utilize publicly available datasets to further enhance precision. FedKD [Gong *et al.*, 2022] makes use of unlabeled, cross-domain public data and requires transferring products of these. To address privacy concerns, it incorporates a quantized and noisy ensemble into local models. Following the ensemble process, it applies distillation to derive the final global model. Notably, by employing a combination of both shared techniques and additional public datasets, the results demonstrate that first ensembling and then distilling yields better outcomes. This is likely because, as mentioned, distillation can be inherently inefficient and may result in some loss of information. Distilling before ensembling might lead to models that are not as accurate prior to aggregation.

Combination with Three Techniques. Dense [Zhang *et al.*, 2022] uses an ensemble of local models uploaded by clients to train a GAN-type generator, which then generates synthetic data samples. The knowledge from the ensemble models is distilled into the global model, which is also trained using the synthetic data to enhance the accuracy. Co-Boosting [Dai *et al.*, 2024] employs a similar generator and also follows the process of ensembling before distillation. However, it extends Dense by introducing a reinforcing approach. This method continuously adjusts the weights of the ensemble model's local models based on synthetic data, and through this process, it finetunes over multiple iterations. Similar to FuseFL [Tang *et al.*, 2024b], although model parameters are uploaded only once, the method undergoes several iterations of optimization, achieving high accuracy. Unlike the previous two methods, in FedCVAE [Heinbaugh *et al.*, 2023], each local client trains an Autoencoder and transmits the local label information and decoder. The server then uses the uploaded client decoder parameters and local label distributions to train a server decoder. Ultimately, synthetic samples from the server

decoders are used to train the global model. FedCAVE-KD employs knowledge distillation of local decoders to train the server decoder, while FedCAVE-Ens ensembles client decoders and trains the server decoder.

To avoid repetition in Section 3.4, we address the privacy issues associated with ensemble methods here, as they often combine with other techniques. For example, using distillation before ensembling can effectively enhance privacy. Conversely, ensembling first and then distilling can achieve security by adding noise to the uploaded local models, similar to parameter learning. Therefore, regardless of how these techniques are combined, we find they all can ensure privacy.

4 Discussions and Future Directions

In this section, having discussed all the current techniques, we note that these methods have successfully implemented one-shot learning, reduced communication overhead, and ensured data privacy. We will summarize the findings based on these methods from the perspective of improving accuracy. Following this, we will present a blueprint for future directions that focuses not only on test accuracy but also considers other aspects, grounded in practical guidance.

4.1 Findings

Prototype learning has a bright future. Parameter learning faces challenges in addressing model heterogeneity and is generally limited by the method's capacity to fully capture the weights in local parameters and discard spurious correlations caused by data heterogeneity, which can restrict accuracy. Consequently, compared to other methods, parameter learning does not demonstrate significant advantages. However, fine-grained approaches such as prototype learning within parameter learning show substantial potential. They enable feature extraction by class, thereby enriching the global model's information. These approaches can be readily integrated with other methods, as seen with FedTMOS, which employs an ensemble method, and FedDISC, which utilizes generative model diffusion. Thus, employing prototype learning or conducting more granular analyses of local models or class features in OFL holds significant promise.

Better Generative Model is needed. Generative models tend to perform better compared to knowledge distillation methods, primarily because they typically operate directly on the data rather than the model. Even when considering data distillation as opposed to model distillation, it becomes challenging to compare with generative methods due to the compression involved in distillation, which limits accuracy. We also observe that the academic community is striving for improved generative models. Therefore, if better generative models become available, they should be utilized in the OFL method. Of course, it is also worth experimenting with improved distillation methods for one-shot Federated Learning.

Adaptive ensemble method shows great potential. Compared to static ensembles, adaptive ensembles achieve better results. Both FENSE [Allouah *et al.*, 2024] and Co-boosting [Dai *et al.*, 2024] consider the differences between global and local models, allowing for a refinement of ensemble weights for local models to enhance accuracy. However,

these dynamic methods still struggle to adjust according to specific data samples. Due to data heterogeneity, the disparity between data samples across different clients can be significant, necessitating varied weight distributions for different samples. IntactOFL [Zeng *et al.*, 2024] significantly improves accuracy using an MoE approach, although it uses a basic generative model like GAN and does not use distillation. Its accuracy surpasses both Dense and Co-boosting. Consequently, a better adaptive ensemble method in OFL that accounts for data samples will likely yield better results.

Uploading once but updating multiple iterations. We found that although OFL restricts the upload to a single round, performing multiple local updates can better leverage the upload. Aside from Co-boosting [Dai *et al.*, 2024], FuseFL [Tang *et al.*, 2024b] updates one block (one or multiple layers) per iteration from the bottom up view, thereby updating from a causal perspective, ultimately achieving excellent results. These findings suggest that a multi-shot approach should be favored over a one-shot approach on the server while keeping the same communication costs.

Aggregating more techniques is preferable. The results from the papers we previously discussed demonstrate that, beyond employing a better model within various techniques, there is no single "silver bullet" for improving one-shot Federated Learning outcomes. Instead, integrating multiple techniques can lead to better results. In detail, incorporating prototype learning when training local models can significantly refine their representation capabilities. Utilizing more fine-grained knowledge distillation when distilling adaptive ensemble models can further enhance the precision and effectiveness of the ensembles. Additionally, training the distilled global model with synthetic datasets generated by superior generative models can improve its robustness and generalization, thereby contributing to enhanced overall performance. Besides these approaches, an ideal method should involve conducting multiple iterations of local updates. In this scenario, the uploaded statistical information can be maximized, leading to the best possible results.

4.2 Future Directions

In addition to the findings we have uncovered that can guide enhancements in the accuracy of OFL methods, this subsection will provide a practical roadmap for the future development of one-shot Federated Learning.

Data-free Requirement. XorMixFL [Shin *et al.*, 2020] and FedCAVE [Heinbaugh *et al.*, 2023] both require the transmission of local clients' data labels, posing a risk of data privacy leakage. On the other hand, methods like One-shot FL [Guha *et al.*, 2019], FedKT [Li *et al.*, 2021b], and FedKD [Gong *et al.*, 2022] depend on additional public datasets; similarly, approaches such as FedPFT [Beitollahi *et al.*, 2024], FedBiP [Chen *et al.*, 2024], and FedDISC [Yang *et al.*, 2024b] utilize foundation models, leveraging extra public datasets as well. However, OFL ideally should be data-free; using additional datasets might lead the global model to learn biased information that does not align with the target dataset, such as certain industrial or biological datasets, which may ultimately reduce accuracy. Therefore, data-free operation should be one of the key objectives for OFL methods.

Scalability for LLMs. Training LLMs in FL with multiple rounds requires significant communication costs because of the enormous model sizes [Ye *et al.*, 2024; Tang *et al.*, 2024a]. OFL has significant potential to reduce this communication cost, thus implementing training or finetuning LLMs across different geo-distributed clients. However, most current experimental approaches for OFL methods primarily focus on models like LeNet, VGGNet, and ResNet, with datasets mostly comprising MNIST, CIFAR, Tiny ImageNet, or simple medical datasets and etc. There is some work trying to merge different finetuned LLMs [Lai *et al.*, 2025] only once. Besides, state-of-the-art OFL methods can accommodate model heterogeneity [Li *et al.*, 2021c; Shen *et al.*, 2025], knowledge distillation between heterogeneous LLMs [Gu *et al.*, 2024] and collaboration between LLMs [Wang *et al.*, 2024a] can be implemented in OFL. In the future, the FL community could explore OFL methods to enable more practical training of LLMs with different parties.

Practical Applications. Based on the experimental considerations mentioned above, besides the LLM field, these methods should be applicable to more practical applications. For instance, combining them with vertical federated learning (VFL) could result in one-shot VFL methods. The OFL approach can be seen in applications like FedISCA [Kang *et al.*, 2023] with biomedical data, FedD3 [Song *et al.*, 2023] focusing on resource-constrained edge environments, and OFL-W3 [Jiang *et al.*, 2024] integrated with blockchain and Web 3.0 technology. Given the one-shot nature of OFL, which eliminates the need for synchronization, it can also be widely utilized in cloud-edge collaborative frameworks. Additionally, there is potential for application in emerging fields like sequential federated learning [Wang *et al.*, 2024b]. These highlight the broad application prospects of OFL.

Advanced Optimizations. Although OFL significantly reduces communication overhead, integrating it with LLMs and cloud-edge collaborative frameworks presents additional challenges. These challenges arise because such models are large and often support numerous services, leading to substantial parameter sizes even in a single round of interaction. Therefore, further optimization is required, such as implementing model compression [Hu *et al.*, 2024] or acceleration techniques [Liu and Zeng, 2024].

In summary, while OFL methods demonstrate tremendous practical potential, improvements should not be limited to enhancing accuracy. Consideration should also be given to the aspects we mentioned. By addressing these considerations and integrating our findings, we outline a clear direction for the future development of OFL methods.

5 Conclusion

In this paper, we focus on the novel distributed machine learning paradigm, one-shot Federated Learning. We provide a detailed analysis of the challenges faced within one-shot Federated Learning, propose an innovative taxonomy, and thoroughly discuss the specifics of various methods based on this

classification. Our paper comprehensively covers all existing OFL literature, comparing the advantages, developmental trajectories, and areas for improvement across different techniques. Based on this analysis, we discuss our findings and offer numerous future directions from a practical standpoint for this rapidly evolving field. This survey aims to consolidate existing knowledge and lay a foundation for future work in the promising one-shot Federated Learning area.

Acknowledgements

References

[Alemdar *et al.*, 2021] Kubra Alemdar, Divashree Varshney, Subhramoy Mohanti, Ufuk Muncuk, and Kaushik Chowdhury. Rfclock: Timing, phase and frequency synchronization for distributed wireless networks. In *MobiCom*, 2021.

[Allouah *et al.*, 2024] Youssef Allouah, Akash Dhasade, et al. Revisiting ensembling in one-shot federated learning. *NeurIPS*, 2024.

[Beitollahi *et al.*, 2024] Mahdi Beitollahi, Alex Bie, et al. Parametric feature transfer: One-shot federated learning with foundation models. *arXiv:2402.01862*, 2024.

[Bonawitz *et al.*, 2017] Keith Bonawitz, Vladimir Ivanov, et al. Practical secure aggregation for privacy-preserving machine learning. In *CCS*, 2017.

[Chen *et al.*, 2024] Haokun Chen, Hang Li, et al. Fedbip: Heterogeneous one-shot federated learning with personalized latent diffusion models. *arXiv:2410.04810*, 2024.

[Cheon *et al.*, 2017] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *Asiacrypt*, 2017.

[Dai *et al.*, 2024] Rong Dai, Yonggang Zhang, Ang Li, Tongliang Liu, Xun Yang, and Bo Han. Enhancing one-shot federated learning through data and ensemble co-boosting. *ICLR*, 2024.

[Dennis *et al.*, 2021] Don Kurian Dennis, Tian Li, and Virginia Smith. Heterogeneity for the win: One-shot federated clustering. In *ICML*, 2021.

[Diao *et al.*, 2021] Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In *ICLR*, 2021.

[Diao *et al.*, 2023] Yiqun Diao, Qinbin Li, and Bingsheng He. Towards addressing label skews in one-shot federated learning. In *ICLR*, 2023.

[Dwork, 2011] Cynthia Dwork. Differential privacy. *Encyclopedia of Cryptography and Security*, 2011.

[Gong *et al.*, 2022] Xuan Gong, Abhishek Sharma, et al. Preserving privacy in federated learning with ensemble cross-domain knowledge distillation. In *AAAI*, 2022.

[Gu *et al.*, 2024] Yuxian Gu, Li Dong, and et al. MiniLLM: Knowledge distillation of large language models. In *ICLR*, 2024.

[Guha *et al.*, 2019] Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.

[Hao *et al.*, 2025] Xiaolei Hao, Li Shen, et al. Exploring one-shot federated learning by model inversion and token relabel with vision transformers, 2025.

[Hasan *et al.*, 2024] Mohsin Hasan, Guojun Zhang, Kaiyang Guo, Xi Chen, and Pascal Poupart. Calibrated one round federated learning with bayesian inference in the predictive space. In *AAAI*, 2024.

[Heinbaugh *et al.*, 2023] Clare Elizabeth Heinbaugh, Emilio Luz-Ricca, et al. Data-free one-shot federated learning under very high statistical heterogeneity. In *ICLR*, 2023.

[Hu *et al.*, 2024] Sixu Hu, Linshan Jiang, and Bingsheng He. Practical hybrid gradient compression for federated learning systems. In *IJCAI*, 2024.

[Humbert *et al.*, 2023] Pierre Humbert, Batiste Le Bars, Aurélien Bellet, and Sylvain Arlot. One-shot federated conformal prediction. In *ICML*, 2023.

[Jacobs *et al.*, 1991] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 1991.

[Jhunjhunwala *et al.*, 2023] Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. Towards a theoretical and practical understanding of one-shot federated learning with fisher information. In *ICML Workshop FL*, 2023.

[Jiang *et al.*, 2024] Linshan Jiang, Moming Duan, Bingsheng He, Yulin Sun, Peishen Yan, Yang Hua, and Tao Song. Ofl-w3: A one-shot federated learning system on web 3.0. *VLDB*, 2024.

[Kairouz *et al.*, 2021] Peter Kairouz, H Brendan McMahan, et al. Advances and open problems in federated learning. *Foundations and trends in machine learning*, 2021.

[Kang *et al.*, 2023] Myeongkyun Kang, Philip Chikontwe, Soopil Kim, et al. One-shot federated learning on medical data using knowledge distillation with image synthesis and client model adaptation. In *MICCAI*, 2023.

[Karimireddy *et al.*, 2020] Sai Praneeth Karimireddy, Satyen Kale, et al. Scaffold: Stochastic controlled averaging for federated learning. In *ICML*, 2020.

[Kasturi *et al.*, 2020] Anirudh Kasturi, Anish Reddy Ellore, and Chittaranjan Hota. Fusion learning: A one shot federated learning. In *ICCS*, 2020.

[Kasturi, 2023] Chittaranjan Kasturi, Anirudh; Hota. Osgan: One-shot distributed learning using generative adversarial networks. *Journal of Supercomputing*, 2023.

[Lai *et al.*, 2025] Kunfeng Lai, Zhenheng Tang, Xinglin Pan, Peijie Dong, and et al. Mediator: Memory-efficient llm merging with less parameter conflicts and uncertainty based routing. *arXiv preprint*, 2025.

[Li *et al.*, 2020] Tian Li, Anit Kumar Sahu, et al. Federated optimization in heterogeneous networks. *MLSys*, 2020.

[Li *et al.*, 2021a] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *CVPR*, 2021.

[Li *et al.*, 2021b] Qinbin Li, Bingsheng He, and Dawn Song. Practical one-shot federated learning for cross-silo setting. *IJCAI*, 2021.

[Li *et al.*, 2021c] Xiaoxiao Li, Meirui JIANG, et al. FedBN: Federated learning on non-IID features via local batch normalization. In *ICLR*, 2021.

[Li *et al.*, 2022] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *ICDE*, 2022.

[Liu and Zeng, 2024] Zhen Liu and Qingmin Zeng. Accelerate one-shot federated learning through ensemble optimization. In *AINIT*, 2024.

[Liu *et al.*, 2024a] Xiang Liu, Liangxi Liu, Feiyang Ye, Yunheng Shen, Xia Li, Linshan Jiang, and Jialin Li. Fedlpa: One-shot federated learning with layer-wise posterior aggregation. In *NeurIPS*, 2024.

[Liu *et al.*, 2024b] Yige Liu, Yiwei Lou, Yang Liu, Yongzhi Cao, and Hanpin Wang. Label leakage in vertical federated learning: A survey. In *IJCAI*, 2024.

[Liu *et al.*, 2024c] Zhi Liu, Xiaohua He, et al. Fedocd: A one-shot federated framework for heterogeneous cross-domain recommendation. In *APWeb-WAIM*, 2024.

[McMahan *et al.*, 2016] H Brendan McMahan, Eider Moore, et al. Federated learning of deep networks using model averaging. *arXiv:1602.05629*, 2016.

[Mora *et al.*, 2024] Alessio Mora, Irene Tenison, Paolo Bellavista, and Irina Rish. Knowledge distillation for federated learning: a practical guide. *IJCAI*, 2024.

[Qi *et al.*, 2025] Shannon How Shi Qi, Jagmohan Chauhan, et al. FedTMOS: Efficient one-shot federated learning with tsetlin machine. In *ICLR*, 2025.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[Rombach *et al.*, 2022] Robin Rombach, Andreas Blattmann, et al. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[Shah and Lau, 2021] Suhail Mohmad Shah and Vincent K. N. Lau. Model compression for communication efficient federated learning. *IEEE TNNLS*, 2021.

[Shen *et al.*, 2025] Lei Shen, Zhenheng Tang, et al. Bridging general and personalized federated learning through selective model integration. In *ICLR*, 2025.

[Shin *et al.*, 2020] MyungJae Shin, Chihoon Hwang, Joongheon Kim, Jihong Park, et al. Xor mixup: Privacy-preserving data augmentation for one-shot federated learning. *arXiv:2006.05148*, 2020.

[Shin *et al.*, 2024] Yujin Shin, Kichang Lee, et al. Effective heterogeneous federated learning via efficient hypernetwork-based weight generation. In *SenSys*, 2024.

[Smith *et al.*, 2017] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. In *NeurIPS*, 2017.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, et al. Prototypical networks for few-shot learning. *NeurIPS*, 2017.

[Song *et al.*, 2023] Rui Song, Dai Liu, et al. Federated learning via decentralized dataset distillation in resource-constrained edge environments. In *IJCNN*, 2023.

[Su *et al.*, 2023] Shangchao Su, Bin Li, and Xiangyang Xue. One-shot federated learning without server-side training. *Neural Networks*, 164:203–215, 2023.

[Tang *et al.*, 2022] Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, and et al. Virtual homogeneity learning: Defending against data heterogeneity in federated learning. In *ICML*, 2022.

[Tang *et al.*, 2024a] Zhenheng Tang, Xueze Kang, and et al. Fusionllm: A decentralized llm training system on geo-distributed gpus with adaptive compression. *arXiv preprint*, 2024.

[Tang *et al.*, 2024b] Zhenheng Tang, Yonggang Zhang, Peijie Dong, Yiu-ming Cheung, Amelie Chi Zhou, Bo Han, and Xiaowen Chu. Fusefl: One-shot federated learning through the lens of causality with progressive model fusion. *NeurIPS*, 2024.

[Wang *et al.*, 2020] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, et al. Federated learning with matched averaging. *ICLR*, 2020.

[Wang *et al.*, 2023] Naibo Wang, Wenjie Feng, Moming Duan, Fusheng Liu, See-Kiong Ng, et al. Data-free diversity-based ensemble selection for one-shot federated learning. *TMLR*, 2023.

[Wang *et al.*, 2024a] Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*, 2024.

[Wang *et al.*, 2024b] Naibo Wang, Yuchen Deng, et al. One-shot sequential federated learning for non-iid data by enhancing local model diversity. In *ACM MM*, 2024.

[Woisetschlager and others, 2024] Herbert Woisetschlager et al. A survey on efficient federated learning methods for foundation model training. *IJCAI*, 2024.

[Yang *et al.*, 2024a] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *IJCV*, 2024.

[Yang *et al.*, 2024b] Mingzhao Yang, Shangchao Su, et al. Exploring one-shot semi-supervised federated learning with pre-trained diffusion models. In *AAAI*, 2024.

[Yang *et al.*, 2024c] Mingzhao Yang, Shangchao Su, Bin Li, and Xiangyang Xue. Feddeo: Description-enhanced one-shot federated learning with diffusion models. *MM*, 2024.

[Ye *et al.*, 2024] Rui Ye, Wenhao Wang, and et al. Openfedllm: Training large language models on decentralized private data via federated learning. In *SIGKDD*, 2024.

[Yurochkin *et al.*, 2019] Mikhail Yurochkin, Mayank Agar-
    wal, et al. Bayesian nonparametric federated learning of
    neural networks. In *ICML*, 2019.

[Zeng *et al.*, 2024] Hui Zeng, Minrui Xu, et al. One-shot-
    but-not-degraded federated learning. *MM*, 2024.

[Zhang *et al.*, 2022] Jie Zhang, Chen Chen, et al. Dense:
    Data-free one-shot federated learning. *NeurIPS*, 2022.

[Zhang *et al.*, 2024a] Guangsheng Zhang, Bo Liu, et al. How
    does a deep learning model architecture impact its pri-
    vacy? a comprehensive study of privacy attacks on cnns
    and transformers. In *USENIX Security*, 2024.

[Zhang *et al.*, 2024b] Junyuan Zhang, Songhua Liu, and
    Xinchao Wang. One-shot federated learning via synthetic
    distiller-distillate communication. *NeurIPS*, 2024.

[Zhou *et al.*, 2020] Yanlin Zhou, George Pu, et al. Distilled
    one-shot federated learning. *arXiv:2009.07999*, 2020.