Who to Trust? Aggregating Client Knowledge in Logit-Based Federated Learning

Viktor Kovalchuk VIKTOR.KOVALCHUK@MBZUAI.AC.AE
Nikita Kotelevskii NIKITA.KOTELEVSKII@MBZUAI.AC.AE
Maxim Panov MAXIM.PANOV@MBZUAI.AC.AE
Samuel Horváth SAMUEL.HORVATH@MBZUAI.AC.AE
Martin Takáč MARTIN.TAKAC@MBZUAI.AC.AE
Mohamed bin Zayed University of Artificial Intelligence, United Arab Emirates

Abstract

Federated learning (FL) usually shares model weights or gradients, which is costly for large models. Logit-based FL reduces this cost by sharing only logits computed on a public proxy dataset. However, aggregating information from heterogeneous clients is still challenging. This paper studies this problem, introduces and compares three logit aggregation methods: simple averaging, uncertainty-weighted averaging, and a learned meta-aggregator. Evaluated on MNIST and CIFAR-10, these methods reduce communication overhead, improve robustness under non-IID data, and achieve accuracy competitive with centralized training.

1. Introduction

Federated learning (FL) enables multiple clients to collaboratively train models without sharing raw data [2, 6, 8, 13, 19]. However, most FL algorithms require transmitting model parameters or gradients [1, 3], leading to high communication costs and degraded performance under data heterogeneity [4, 9, 14, 16, 18]. To address this, several works explore logit-based Federated Distillation, where clients share model outputs instead of parameters [11, 17, 20].

In this work, we study a logit-based federated learning (FL) scheme in which clients, each trained on datasets with partial class coverage, exchange only logits evaluated on a shared, unlabeled public dataset. The aggregated logits are then used as soft targets to refine local models, enabling knowledge transfer without sharing gradients or model weights.

Contributions. Our work makes three main contributions: (1) building on prior work on logit-based training, we introduce a communication-efficient federated distillation techniques tailored to heterogeneous client settings, which entirely avoids transmitting gradients or model parameters; (2) we propose three strategies for logit aggregation – simple averaging, uncertainty-weighted averaging with Gaussian Mixture Models, and a learned Meta-Model Aggregator; and (3) we provide empirical evidence on MNIST and CIFAR-10 showing that our method achieves robust performance under significant data heterogeneity.

2. Related Work

The standard FedAvg algorithm [13] averages client *weight updates* (or equivalently model parameters) across communication rounds, but suffers under heterogeneous data distributions [7]. Logit-based methods such as FedMD [11] introduce knowledge distillation using a shared public dataset, where models exchange predictions instead of weights. Ensemble distillation approaches [12] further highlight the potential of aggregating client logits to improve generalization. Our work extends these ideas using an unlabeled shared dataset and suggests new aggregation techniques.

3. Proposed Method

We consider a particular distribution shift across clients – *label-distribution shift with support mismatch*. Specifically, we consider M clients and assume that each client i observes only a subset of classes $\mathcal{C}_i \subseteq \{1, \ldots, C\}$ and has a different label prior $p_i(y)$. Additionally, for simplicity, we assume that $|\mathcal{C}_i| = k$ (fixed constant) for all i. Moreover, all clients have access to a shared public dataset $\mathcal{D}_{pub}$ containing unlabeled samples from all C classes.

We summarize the overall workflow of the proposed method in Algorithm 1. Each communication round consists of three main stages: (i) local training on private client datasets, (ii) generation and aggregation of client logits on the shared public dataset, and (iii) client refinement using the aggregated logits as soft targets. This procedure enables knowledge transfer across clients without sharing raw data, model parameters, or gradients.

Algorithm 1 Logit-Based Federated Learning Workflow

```
Require: Client datasets $\{\mathcal{D}_i\}_{i=1}^M$, public dataset $\mathcal{D}_{pub}$, number of rounds $R$
Ensure: Trained client models $\{f_i\}_{i=1}^M$
for $r = 1$ to $R$ do
    for each client $i = 1, \ldots, M$ do
        Train local model $f_i$ on private dataset $\mathcal{D}_i$
        Evaluate $f_i$ on $\mathcal{D}_{pub}$ to obtain logits $z_i(x)$ for each $x \in \mathcal{D}_{pub}$
    end for
    for each sample $x \in \mathcal{D}_{pub}$ do
        $z(x) \leftarrow$ Aggregation($\{z_i(x)\}_{i=1}^M$)
    end for
    for each client $i = 1, \ldots, M$ do
        Retrain $f_i$ on $\mathcal{D}_{pub}$ using aggregated logits $z(x)$ as soft targets
    end for
end for
```

3.1. Aggregation Methods

In communication-efficient federated learning, instead of directly averaging model parameters as in FedAvg [13], we focus on *logit aggregation*: each client computes class logits on a shared dataset, and these are combined at the server to form supervisory signals for further training. We consider three aggregation strategies.

3.1.1. Simple Averaging of Logits

The most straightforward method is to average the logits produced by all local models. Let M be the number of clients, x an input sample, and $f_i(x) \in \mathbb{R}^C$ the logits of model i with C classes. The aggregated logits are

$$\bar{z}(x) = \frac{1}{M} \sum_{i=1}^{M} f_i(x), \tag{1}$$

and the corresponding soft labels are obtained via the softmax function:

$$p(x) = \operatorname{softmax}\big(\bar{z}(x)\big). \tag{2}$$

This simple averaging allows all clients to benefit from the combined knowledge of peers, while requiring only logits to be shared, reducing communication cost. However, this simple aggregation treats predictions from each model f_i equally important, and ignores the situation when the input object x belongs to the class out of the training split D_i. To address this issue, we propose more sophisticated aggregation ideas.

3.1.2. AGGREGATION VIA AN AUXILIARY MODEL

Instead of fixed averaging, we can train a *separate Meta-Model Aggregator* that learns how to combine logits. Each client produces logits $f_i(x)$, which are concatenated into a feature vector:

$$h(x) = \left[f_1(x)^T, f_2(x)^T, \dots, f_M(x)^T\right] \in \mathbb{R}^{M \cdot C}. \tag{3}$$

The aggregator $A(\cdot)$ (e.g., a small neural network, random forest, or gradient boosting model) maps this concatenated vector to predicted labels:

$$A\colon \mathbb{R}^{M \cdot C} \to \mathbb{R}^{C}, \qquad \hat{y}(x) = \arg\max\, A\big(h(x)\big). \tag{4}$$

This approach is more flexible than averaging, since the aggregator can learn to *weight clients differently depending on the input*. However, the aggregator must be trained on a common labeled dataset, which increases the computational cost. Another limitation is that in case of missing information from a client(s), or in case of adding a new one, the whole model should be retrained. For the experiments, the labeled dataset is constructed from images available in local datasets, without incorporating any additional information for the models.

3.1.3. UNCERTAINTY WEIGHED AVERAGING (UWA)

Recap, that we consider a label shift data heterogeneity. Under the label-shift assumption that the class-conditionals $p(x \mid y)$ are shared, the differing priors imply different feature marginals $p_i(x) = \sum_y p(x \mid y) p_i(y)$ across clients. Thus, the distributions of model logits also differ from client to client. We consider each client's *personalized (local) logit distribution* to account for this heterogeneity.

Specifically, for client i, recap that $f_i(x) \in \mathbb{R}^C$ denotes the logits produced by its local model. Using a local validation set restricted to the classes in $\mathcal{C}_i$, we fit a Gaussian mixture density over $f_i(x)$, with the number of components set to $|\mathcal{C}_i|$ (one component for a particular class). For a test input x, we compute a confidence score:

$$\ell_i(x) := \log \left(\frac{1}{|\mathcal{C}_i|} \sum_{k \in \mathcal{C}_i} (2\pi)^{-C/2} \prod_{d=1}^{C} \sigma_{i,k,d}^{-1} \exp\left[-\tfrac{1}{2} \sum_{d=1}^{C} \frac{\big(f_{i,d}(x) - \mu_{i,k,d}\big)^2}{\sigma_{i,k,d}^2} \right] \right), \tag{5}$$

where we assume a mean-field approximation to each component and uniform component weights.

These log-likelihoods are further normalized into weights using a softmax:

$$w_i(x) = \frac{\exp\big(\ell_i(x)\big)}{\sum_{j=1}^{M} \exp\big(\ell_j(x)\big)}. \tag{6}$$

The final aggregated logits are then a *confidence-weighted average*:

$$z(x) = \sum_{i=1}^{M} w_i(x) f_i(x), \quad p(x) = \mathrm{softmax}\big(z(x)\big). \tag{7}$$

This procedure generalizes simple averaging by down-weighting clients on unlikely inputs under their logit distributions. While logit-density scores have been used for out-of-distribution detection [15], here we adapt them to *federated* aggregation under label-distribution shift. In comparison with previous method, this one is robust for adding/missing clients, since we only need to train local model, not to train the whole thing.

4. Experiments

4.1. Setup

We evaluate the proposed logit aggregation methods on two widely used benchmark datasets: MNIST and CIFAR-10. To simulate heterogeneous settings, each client is assigned a local dataset containing only a subset of classes, with $k \in \{2, 5, 8\}$ classes per client. The federation consists of a total of 20 clients, each training independently on its restricted class distribution. Training proceeds in rounds, where clients compute logits on a shared unlabeled dataset, exchange them for aggregation, and then refine their local models with the aggregated soft targets.

4.2. Models

For MNIST, we adopt LeNet architectures [10] as local client models. For CIFAR-10, ResNet-18 architectures [5] are used. Our method does not require all clients to use the same architecture – the aggregation relies only on logits, and heterogeneous client models could also be supported.

4.3. Evaluation Metrics

We report test accuracy averaged across all client models, which reflects both generalization performance and fairness across heterogeneous client distributions. The *Fully Informed Reference* denotes the accuracy achieved by training the same model architecture under standard supervised learning on a dataset containing all classes. To ensure comparability, the reference dataset is matched in size to that available to each client (public data plus local dataset), so that the reference model has been trained on the same number of images as local clients. The Fully Informed Reference for MNIST reaches $97.97\% \pm 0.18\%$, while for CIFAR-10 it reaches $84.53\% \pm 0.45\%$.

4.4. Results

The results are summarized in Figure 1. Simple averaging performs poorly under *high* client heterogeneity (e.g., $k = 2$ classes per client), but as heterogeneity decreases (larger k), all methods move closer to the Fully-Informed Reference.

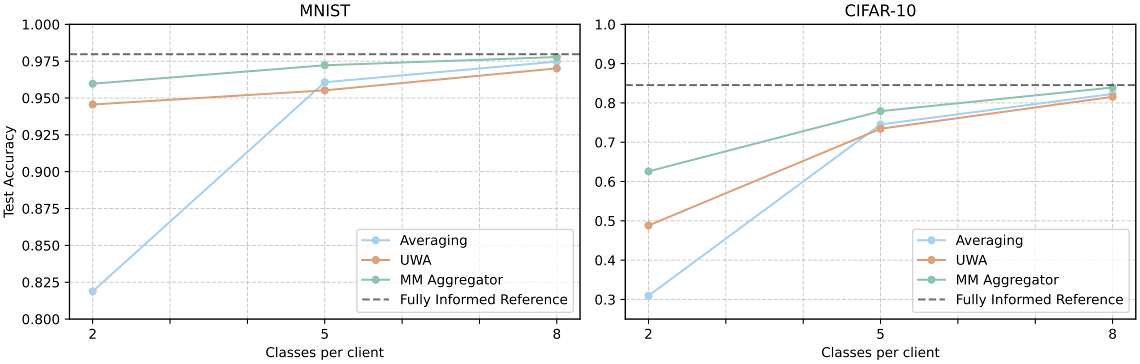

Figure 1: Accuracy of different aggregation methods compared to the Fully Informed Reference on MNIST and CIFAR-10.

Unlike averaging, **UWA** provides substantial gains under strong heterogeneity: with $k = 2$, it markedly improves over simple averaging on both MNIST and CIFAR-10. As the number of classes per client increases ($k = 5, 8$), client predictions become more confident and the learned weights tend toward a near-uniform distribution, making UWA increasingly similar to averaging (and occasionally slightly below it).

MM Aggregator consistently yields the best performance across heterogeneity levels, narrowing the gap to the Fully-Informed Reference most effectively (see Table 1 for more details).

5. Conclusion

We studied federated distillation under data heterogeneity (label shift) through a logit-based framework, where clients exchange prediction logits instead of model parameters. This approach eliminates the need for gradient or weight transfer, substantially reducing communication costs. We investigated three aggregation strategies. Specifically, simple averaging, uncertainty-weighted averaging, and a meta-model aggregator. We evaluated them on MNIST and CIFAR-10 with varying levels of class imbalance across clients.

Our experiments show that naive averaging is highly sensitive to non-IID data, while more informed strategies, particularly the meta-model aggregator, achieve significantly higher robustness and accuracy, in approaching fully informed reference, but suffer from higher computational cost and non-flexibility in case of changeable number of clients. These findings demonstrate that the choice of aggregation strategy is crucial for effective logit-based federated learning.

Acknowledgements

Viktor Kovalchuk began exploring this research topic during the Summer School on Federated Learning at the Yandex School of Data Analysis in 2025. The initial development of this work benefited from fruitful discussions with Dariya Shevyakova, Jacob Denisov, Evgeny Uskov, and was guided by the mentorship of Aleksandr Beznosikov, Dmitry Bylinkin, Daniil Medyakov.

The authors would also like to thank the organizers and participants of the Yandex School of Data Analysis summer program for creating an inspiring research environment.

References

[1] Artem Agafonov, Dmitry Kamzolov, Rachael Tappenden, Alexander Gasnikov, and Martin Takáč. Flecs: A federated learning second-order framework via compression and sketching. *Optimization Methods and Software*, pages 1–27, 2025.

[2] Ekaterina Borodich, Aleksandr Beznosikov, Abdurakhmon Sadiev, Vadim Sushko, Nikolay Savelyev, Martin Takác, and Alexander Gasnikov. Decentralized personalized federated minmax problems. *arXiv preprint arXiv:2106.07289*, 2021.

[3] Yury Demidovich, Petr Ostroukhov, Grigory Malinovsky, Samuel Horváth, Martin Takáč, Peter Richtárik, and Eduard Gorbunov. Methods with local steps and random reshuffling for generally smooth non-convex federated optimization. *ICLR 2025*, 2025.

[4] Dmitrii Feoktistov, Igor Ignashin, Andrey Veprikov, Nikita Borovko, Alexander Bogdanov, Savelii Chezhegov, and Aleksandr Beznosikov. Aligning distributionally robust optimization with practical deep learning needs. *arXiv preprint arXiv:2508.16734*, 2025.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

[7] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.

[8] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[9] Nikita Kotelevskii, Samuel Horváth, Karthik Nandakumar, Martin Takáč, and Maxim Panov. Dirichlet-based uncertainty quantification for personalized federated learning with improved posterior networks. *arXiv preprint arXiv:2312.11230*, 2023.

[10] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[11] Daoyuan Li and Junpu Wang. Fedmd: Heterogeneous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

[12] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[13] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282, 2017.

[14] Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences. *Optimization Methods and Software*, pages 1–16, 2024.

[15] Jishnu Mukhoti, Andreas Kirsch, Joost Van Amersfoort, Philip HS Torr, and Yarin Gal. Deep deterministic uncertainty: A new simple baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24384–24394, 2023.

[16] Abdurakhmon Sadiev, Ekaterina Borodich, Aleksandr Beznosikov, Darina Dvinskikh, Saveliy Chezhegov, Rachael Tappenden, Martin Takáč, and Alexander Gasnikov. Decentralized personalized federated learning: Lower bounds and optimal algorithm for all personalization modes. *EURO Journal on Computational Optimization*, 10:100041, 2022.

[17] Jiawei Shao, Fangzhao Wu, and Jun Zhang. Selective knowledge sharing for privacy-preserving federated distillation without a good teacher. *Nature Communications*, 15(1):349, 2024.

[18] Nazarii Tupitsa, Samuel Horváth, Martin Takáč, and Eduard Gorbunov. Federated learning can find friends that are advantageous. *arXiv preprint arXiv:2402.05050*, 2024.

[19] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

[20] Guoyizhe Wei and Xiu Li. Knowledge lock: Overcoming catastrophic forgetting in federated learning. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 601–612. Springer, 2022.

Appendix A. Results Details

Table 1: Test accuracy of different aggregation methods on MNIST and CIFAR-10 with varying number of classes per client.

Dataset	Method	2 classes	5 classes	8 classes
MNIST	Average	0.8188	0.9607	0.9747
	UWA	0.9454	0.9552	0.9701
	MM Aggregator	**0.9597**	**0.9722**	**0.9778**
CIFAR-10	Average	0.3093	0.7452	0.8230
	UWA	0.4881	0.7343	0.8155
	MM Aggregator	**0.6258**	**0.7791**	**0.8392**

A.1. Training Details

All models are trained using the Adam optimizer with momentum 0.9 and a learning rate of 0.001, with a batch size of 128 and Cross-Entropy Loss. Each communication round consists of training on client data, followed by training on aggregated logits. For MNIST, the first round uses 10 epochs for each stage, while subsequent rounds use a single epoch per stage. For CIFAR-10, the first round uses 20 epochs for each stage, after which both are reduced to 5 epochs per round. Unless otherwise specified, training is performed for 50 communication rounds, with convergence typically reached within the first 10 rounds.

A.2. Dataset Details

Each client is assigned a private dataset of $10,000$ images. Since clients may share common classes, their private datasets are not disjoint and can contain overlapping samples. In addition, we assume access to a public dataset of $5,000$ images, which is used for logit aggregation and knowledge transfer. Furthermore, an additional set of $3,000$ images is reserved specifically for training the auxiliary model in MM Aggregator. In order to avoid adding new information to model the additional dataset is constructed using images from private datasets.

Code Availability. Implementations of all aggregation methods and experiments are available at `https://github.com/kovalchuk026/fd_aggregators`.