ON THE EFFECTIVENESS OF ONE-SHOT FEDERATED ENSEMBLES IN HETEROGENEOUS CROSS-SILO SETTINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) is a popular approach for training machine learning models on decentralized data. For communication efficiency, one-shot FL trades the iterative exchange of models between clients and the FL server for one single round of communication. However, one-shot FL does not perform as well as iterative FL, and struggles under high data heterogeneity. While ensembles have repeatedly appeared as strong contenders in one-shot FL literature, their full potential is still under-explored. In this work, we extensively examine federated ensembles across the heterogeneity spectrum, in conjunction with various aggregation functions and a specific focus on cross-silo settings. Our experiments reveal that an aggregator based on a shallow neural network can significantly boost the performance of ensembles under high data heterogeneity. Through comprehensive evaluations on the CIFAR-10, SVHN and the cross-silo healthcare FLamby benchmark, we show that federated ensembles not only achieve up to 26% higher accuracy over current one-shot methods but can also match the performance of iterative FL under high data heterogeneity, all while being up to $9.1\times$ more efficient in terms of communication due to their one-shot nature.

1 INTRODUCTION

FL is a widely adopted distributed machine learning (ML) approach, enabling clients to *collaboratively train* a common model over their collective data without sharing raw data with a central server (McMahan et al., 2017). In each round of collaborative training, clients perform local model updates and send the resulting model to the central server. The server then aggregates the models and disseminates the updated global model to clients for the next round. As clients never share raw data during training, FL has been used in many applications where data privacy is critical, such as recommender systems (Hartmann et al., 2019) and medicine (Li et al., 2019). However, typical FL training tasks take thousands of communication rounds and require coordination by the central server (Bonawitz et al., 2019), which results in major communication costs (Kairouz et al., 2021).

One-shot FL addresses the communication challenges in FL by restricting the exchange of models to a single round, often by forming an ensemble of locally-trained client models at the server (Guha et al., 2019). In fact, ensembles have recurrently appeared as strong competitors in the field of one-shot FL (Guha et al., 2019; Yurochkin et al., 2019; Chen & Chao, 2021; Heinbaugh et al., 2023). However, their full potential is still under-explored, with existing work only considering simple averaging to aggregate the predictions. While other one-shot FL methods have been proposed (Yurochkin et al., 2019; Li et al., 2021), their performance is typically only evaluated against one-round FL algorithms (*e.g.*, FEDAVG). Yet, one-shot methods considerably underperform when compared to *iterative* FL. Likewise, data heterogeneity is only rarely considered in one-shot FL (Heinbaugh et al., 2023).

The potential of ensembles is even greater in *cross-silo* settings. While *cross-device* is the most common use case in FL, the numerous clients with limited data yield *weak classifiers* (Kearns, 1988), unable to produce acceptable accuracy (Lin et al., 2020), either individually or as part of an ensemble. In contrast, cross-silo settings (Kairouz et al., 2021), where a handful of large clients (*e.g.*, banks and hospitals) individually hold sufficiently large amounts of data, produce performant classifiers. At the same time, cross-silo settings are highly heterogeneous, hampering the performance

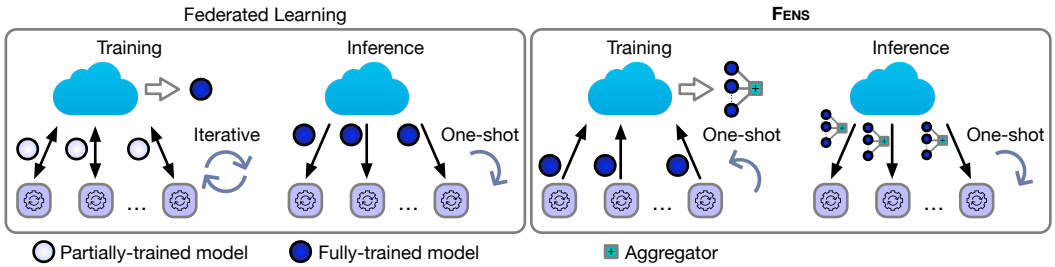

Figure 1: FENS in comparison to Federated Learning.

of FL (Ogier du Terrail et al., 2022). Interestingly, in such scenarios, putting together specialized classifiers in an ensemble might provide superior accuracy than iteratively training a single global model over heterogeneous data.

In the light of above, we thoroughly investigate Federated Ensembles (FENS) in one-shot FL (Figure 1) with various aggregation functions and varying statistical heterogeneity. Focusing on the cross-silo setting, we make the following contributions.

Contributions. *(i)* Our study of different aggregation functions unveils an aggregator based on a shallow neural network, which can significantly boost the performance of ensembles under high data heterogeneity. *(ii)* We perform a comparative study with several state-of-the-art iterative FL algorithms over the CIFAR-10 dataset and conclude that FENS can match their performance in the face of high data heterogeneity, while being up to $9.1\times$ more efficient in communication cost. *(iii)* We extend our analysis to two one-shot baselines — FEDKD (Gong et al., 2022) and FEDCVAE-ENS (Heinbaugh et al., 2023) — and show that FENS can achieve up to 26% higher accuracy under high data heterogeneity. *(iv)* We obtain insights into FENS by analyzing its performance according to the local dataset size. Our experiments on the SVHN datatset reaveal that, for large enough dataset sizes (which is usually the case in cross-silo settings), FENS can match the performance of iterative FL under high heterogeneity. *(v)* Lastly, we evaluate FENS on three tasks from the FLamby benchmark (Ogier du Terrail et al., 2022), a realistic cross-silo FL dataset for healthcare applications. Our results reveal that FENS achieves superior performance over one-shot FEDAVG and one-shot FEDPROX while also matching iterative FL.

2 RELATED WORK

Statistical heterogeneity in FL. Following FEDAVG (McMahan et al., 2017), several algorithms have been developed to address the challenge of data heterogeneity in FL. FEDPROX (Li et al., 2020) constrains local updates through a proximal term, while SCAFFOLD (Karimireddy et al., 2020) uses control variates to correct the client drift. FEDNOVA (Wang et al., 2020b), in turn, combines normalized gradients to tackle heterogeneous updates, whereas FEDYOGI and FEDADAM extend the adaptive optimizers from traditional ML to federated settings (Reddi et al., 2021), delivering superior performance. Other approaches tackle heterogeneity with methods that cluster statistically similar clients (Briggs et al., 2020; Liu et al., 2022) or with advanced client selection strategies (Tang et al., 2022). We remark that all these methods are designed for standard FL and therefore not feasible in one-shot FL. Nevertheless, we perform extensive empirical evaluations and show that FENS can nearly match the performance of several of the above algorithms under high statistical heterogeneity, despite its one-shot nature.

One-shot FL. In the context of cross-device FL, Guha et al. (2019) proposed one-shot FL, where communication is restricted to a single round. They presented two distinct approaches: *(i)* heuristic selection techniques to determine which clients are included in the final ensemble; and *(ii)* knowledge distillation (KD), which requires an auxiliary dataset for ensemble aggregation into another model at the server. After that, other methods for one-shot FL based on KD were proposed (Li et al., 2021; Gong et al., 2022). However, the additional dataset required by KD methods may not be easy to find (Zhu et al., 2021), since it must be large, publicly available, and somewhat similar to the local datasets of clients. To circumvent this issue,Heinbaugh et al. (2023) proposed a data-free one-shot FL using conditional variational autoencoders (VAEs). In their approach, called FEDCVAE-ENS, clients locally train VAEs and upload the decoder to the server, which then generates synthetic samples

from the received decoders and trains a classifier model. Apart from the above, other approaches in one-shot FL either do not fully consider statistical heterogeneity, *i.e.*, non independent and identically distributed (non-IID) data (Shin et al., 2020; Li et al., 2021), or face difficulties under high data heterogeneity (Zhou et al., 2020; Zhang et al., 2021).

Another line of research in FL that tackles the problem of aggregating fully trained client local models (Yurochkin et al., 2019; Wang et al., 2020a), also partly falls in the one-shot regime. Yurochkin et al. (2019) developed PFNM, a Bayesian non-parametric approach for fully-connected neural network architectures that matches neurons in different client models before averaging. Wang et al. (2020a) proposed FEDMA, which extends PFNM to convolutional neural networkss (CNNs) and LSTMs. However, FEDMA performs iterative layer-wise matching, requiring at least as many communication rounds as the network depth, hence not truly being one-shot.

Ensembles in FL. Ensembles have been previously studied in FL for a variety of different goals. Lin et al. (2020) propose FEDDF that performs robust model fusion of client ensembles to support model heterogeneity. The FEDBE algorithm (Chen & Chao, 2021) uses Bayesian Model Ensemble to aggregate client local models, improving over traditional weighted average. Hamer et al. (2020) propose the FEDBOOST algorithm, where the server has a collection of q pre-trained predictors and aims to learn weights $\alpha_1, \alpha_2, \ldots, \alpha_q$ from the federated network for constructing an ensemble $\sum_{k=1}^{q} \alpha_k h_k$ that minimizes the global loss. However, these works are designed for standard FL and rely on substantial iterative communication. In the decentralized edge setting, Shlezinger et al. (2021) show that collaborative inference via neighbor averaging can achieve higher accuracy over local inference alone. However, they assume a setting where clients can exchange query data during inference and consider only IID data replicated on all edge devices.

3 DESCRIPTION OF FENS ALGORITHM

Consider a classification task with input space $\mathcal{X}$ and output space $\mathcal{Y}$. In this paper, we study a setting comprising M clients and a trusted central server. Each client i holds a local dataset $\mathcal{D}_i$ in $\mathcal{X} \times \mathcal{Y}$, which is typically different across clients, *i.e.*, data is heterogeneous. Given a parametric model $h_\theta : \mathcal{X} \to \mathcal{Z}$, each data point $(x, y) \in \mathcal{X} \times \mathcal{Y}$ incurs a loss of $\ell(h_\theta(x), y)$ where $\ell : \mathcal{Z} \times \mathcal{Y} \to \mathbb{R}$. Denoting by $\Theta \subseteq \mathbb{R}^d$ the set of possible parameters for h_θ, the objective of a machine learning algorithm is to solve the empirical risk minimization (ERM) problem defined as

$$\min_{\theta \in \Theta} \quad \frac{1}{|\hat{\mathcal{D}}|} \sum_{(x,y) \in \hat{\mathcal{D}}} \ell\left(h_\theta(x), y\right) \tag{1}$$

where $\hat{\mathcal{D}} := \bigcup_{i \in [M]} \mathcal{D}_i$ is the union of the local datasets held by all clients.

3.1 FEDERATED LEARNING (FL)

FL algorithms, such as FedAvg (McMahan et al., 2017), can be seen as gradient-based optimization methods approximating a minimizer for the ERM. The success of these methods is mainly due to their capacity to solve the ERM problem (Equation (1)), while allowing the data to remain distributed. Essentially, an FL algorithm is an iterative method where the server updates a global parameter thanks to local computations made by the clients. Specifically, the server starts by initializing the model parameter to θ_0. Then it proceeds in T rounds. At each round, the server multicasts the current model parameter θ_t to a set of clients. Then, each client updates its local parameters using local training data, *i.e.*, $\theta_{t+1}^{(i)} \leftarrow \mathrm{ClientLocalUpdate}(\theta_t; \mathcal{D}_i)$. The round finishes with each client sending their updated model $\theta_{t+1}^{(i)}$ to the server, which then updates the global parameters as

$$\theta_{t+1} \leftarrow \frac{1}{M} \sum_{i \in [M]} \theta_{t+1}^{(i)}. \tag{2}$$

3.2 FEDERATED ENSEMBLES (FENS)

In FENS, the iterative parameter exchanges of FL are replaced by a *one-shot* communication of locally trained models in the following procedure:

- **Local training:** Each client i performs a full local training procedure to solve their own ERM problem, *i.e.*, they solve Equation (1) where $\hat{\mathcal{D}}$ is replaced by $\mathcal{D}_i$.
- **One-shot communication:** After computing an approximate solution $\theta^{(i)}$ to their local ERM problem, each client i sends $\theta^{(i)}$ to the server.
- **Global model:** Upon receiving the local parameter $\theta^{(i)}$, corresponding to parametric model $\pi_i \coloneqq h_{\theta^{(i)}}$, the server builds a global model $\pi \colon \mathcal{X} \to \mathcal{Z}$ of the form

$$\pi = f(\pi_1, \ldots, \pi_M) \tag{3}$$

where $f : \mathcal{Z}^M \to \mathcal{Z}$ is an aggregation function.

Accordingly, the design of FENS consists in choosing the aggregation function f in the global model computation (Equation (3)). If proxy data $\mathcal{D}_{\text{proxy}}$ is available at the server, this function f can be *learned* on $\mathcal{D}_{\text{proxy}}$ using the local models $\pi_1, \ldots, \pi_M$. Otherwise, if such data is unavailable, f can be chosen *independently* of the local models, *e.g.*, by averaging. In Section 3.3, we give concrete examples of aggregation methods.

Communication cost analysis. To compare the communication cost of FENS against that of iterative FL, we take into account both training and inference[1] costs. For FENS, the local models $(\pi_1, \ldots, \pi_M)$ along with the aggregation f constitute the global model (Figure 1). The cost of uploading the trained models to the server in one shot entails the training communication costs for FENS. For local inference, the models and aggregation are sent back to clients, which constitute the communication costs associated with inference.

Table 1: Comparison of communication costs between FENS and iterative FL for a total of M clients. Given that in cross-silo FL $M \ll T$, FENS incurs significantly lower total communication w.r.t. FL.

PHASE	FL	FENS
Training	$2dMT$	dM
Inference	dM	$\lvert f\rvert M + dM^2$
Total	$\mathcal{O}(dMT)$	$\mathcal{O}(dM^2)$

In contrast, the training phase of FL consists of several rounds of model exchange with the server, each of significant size, depending on the model dimension d. Indeed, over T rounds of communication, the server and clients exchange $2 \times T \times M$ messages, each of size d. However, FL inference cost is cheaper, since only the single final model needs to be shipped to each client. Essentially, FENS trades the huge training communication cost of FL for a slightly larger inference cost. We detail these costs in Table 1. Symbols d and $|f|$ indicate the size of the local models and aggregation model, respectively. T is the total number of communication rounds for FL.

3.3 FENS AGGREGATION METHODS

For performing the ensemble aggregation, we have different choices for f in Equation (3). We classify these methods into *trainable* and *non-trainable* methods and evaluate them in Section 4. For simplicity, we consider $\mathcal{Y} = [C]$, *i.e.*, multi-class classification with C classes, and $\mathcal{Z}$ to be the probability simplex over $\mathcal{Y}$. Thus, each model π_i maps a data point to a probability distribution over $[C]$, *i.e.*, a C-dimensional vector.

Trainable methods. Trainable methods correspond to parametric aggregations trained on proxy data available at the server, using the local models $\pi_1, \ldots, \pi_M$. The proxy data is commonly assumed to be available at the server and is used by FL model engineers to bootstrap tasks or perform initial hyperparameter exploration (Bonawitz et al., 2019). This dataset could be derived from publicly available data, a portion of clients' data that is not privacy-sensitive, or be synthetically generated (Kairouz et al., 2021). Trainable methods can be generically formulated as the search for parametric model $f_{\boldsymbol{\lambda}}$ which solves the ERM Problem (Equation (4)) for $\boldsymbol{\lambda}$ in parameter space $\Lambda \subseteq \mathbb{R}^q$ on proxy dataset $\mathcal{D}_{\text{proxy}}$:

$$\min_{\boldsymbol{\lambda} \in \Lambda} \quad \frac{1}{|\mathcal{D}_{\text{proxy}}|} \sum_{(x,y) \in \mathcal{D}_{\text{proxy}}} \ell\left(f_{\boldsymbol{\lambda}}(\pi_1(x), \ldots, \pi_M(x)), y\right). \tag{4}$$

[1] In this context, we use the term "inference" to refer to the costs incurred when deploying FENS or FL after the training phase is complete. This should not be confused with the cost associated with performing a single inference, *i.e.*, labeling unknown data.

As such, the learned aggregation function $f_{\boldsymbol{\lambda}}$ can take the form of a neural network, or a linear model as in (Hamer et al., 2020). Recall that the communication cost of FENS in the deployment phase is driven by $|f_{\boldsymbol{\lambda}}| = q$. In fact, we show that a shallow neural network suffices for $f_{\boldsymbol{\lambda}}$ for several tasks. As a consequence, q can be significantly smaller than d. The additional communication cost of transferring the aggregator is therefore minimal. We also explore trainable methods inspired from voting theory such as *polychotomous voting* (Ben-Yashar & Paroush, 2001) (Appendix C.2).

Non-trainable methods. Non-trainable methods correspond to static aggregation rules like averaging, median, weighted averaging *etc*., commonly used in ensemble learning (Polikar, 2006; Ganaie et al., 2022). Of particular interest to us is the *weighted averaging* method (Gong et al., 2022), which we generically formulate as

$$f(\pi_1, \ldots, \pi_M; \boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_M) = \sum_{i=1}^{M} \boldsymbol{\lambda}_i \odot \pi_i \tag{5}$$

where $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_M \in [0,1]^C$ are weight vectors and $\odot$ denotes coordinate-wise product. Plain averaging corresponds to $\boldsymbol{\lambda}_i = [\frac{1}{M}, \ldots, \frac{1}{M}]$. In particular, this method can use the number of samples of a given class that a client possesses in order to weigh the probabilities. However, this requires each client to share its label distribution, which may be privacy-invasive in some applications.

4 EXPERIMENTS

We split our evaluation into the following sections: *(i)* the CIFAR-10 dataset (Krizhevsky et al., 2014) to evaluate FENS in comparison to FL algorithms (Section 4.1); *(ii)* the SVHN (Netzer et al., 2011) dataset to assess the impact of data size w.r.t. data heterogeneity in FENS and FEDAVG (Section 4.2); and *(iii)* the realistic cross-silo FLamby (Ogier du Terrail et al., 2022) benchmark (Section 4.3).

4.1 CIFAR-10

4.1.1 DATA PARTITIONING

For all experiments in this section, we consider a setting with 20 clients on the CIFAR-10 dataset. Its original training and testing sets consist of 50 000 and 10 000 samples, respectively. We split the original training set into 90-10% to respectively obtain $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{proxy}}$, whereas the testing set is split (50-50%) into $\mathcal{D}_{\text{test}}$ and $\mathcal{D}_{\text{val}}$. We use $\mathcal{D}_{\text{val}}$ to tune hyperparameters and always report the accuracy on $\mathcal{D}_{\text{test}}$. To obtain the client datasets, we partition $\mathcal{D}_{\text{train}}$ using the Dirichlet distribution $\texttt{Dir}_{20}(\alpha)$, as done in previous work (Wang et al., 2020b; Gong et al., 2022). The parameter α determines the degree of heterogeneity, with lower values indicating non-IID data distributions (see Appendix A, Figure 5). The $\mathcal{D}_{\text{proxy}}$ split is located at the server and used by FENS to train the aggregator. For FL algorithms, we partition the entire training set ($\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{proxy}}$) across clients. This enables a fair comparison between FL and FENS, both seeing equal amounts of training data. Similarly, non-trainable FENS methods (Section 3.3) also partition $\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{proxy}}$ across clients.

4.1.2 SETUP AND HYPERPARAMETERS

We use ResNet-8 (He et al., 2016) as the model and a fixed batch size of 16 across all our experiments. Other parameters for each setup are described below.

FENS methods. Each client in FENS performs local training for 250 epochs using SGD as the local optimizer. The learning rate is set to 2.5×10^{-3} and decayed using Cosine Annealing. For the neural network (NN) aggregation method, we use a simple multilayer perceptron with 1 hidden layer comprising of 120 units each, using ReLu activations and a final softmax output layer. The NN aggregator is trained using the ADAM optimizer for 300 epochs with a learning rate of 5×10^{-5}. It further adopts an exponential decay with the parameter $\gamma = 0.999$. Parameters for other aggregation methods presented in Section 3.3 are described in Appendix B.1.

Baseline FL algorithms. For comparison with FL, we consider 6 algorithms, including the standard FEDAVG (McMahan et al., 2017); the heterogeneity-robust FEDPROX (Li et al., 2020), FEDNOVA (Wang et al., 2020b) and SCAFFOLD (Karimireddy et al., 2020); and the adaptive optimizer-based FEDYOGI and FEDADAM (Reddi et al., 2021). We tune learning rates for each algorithm

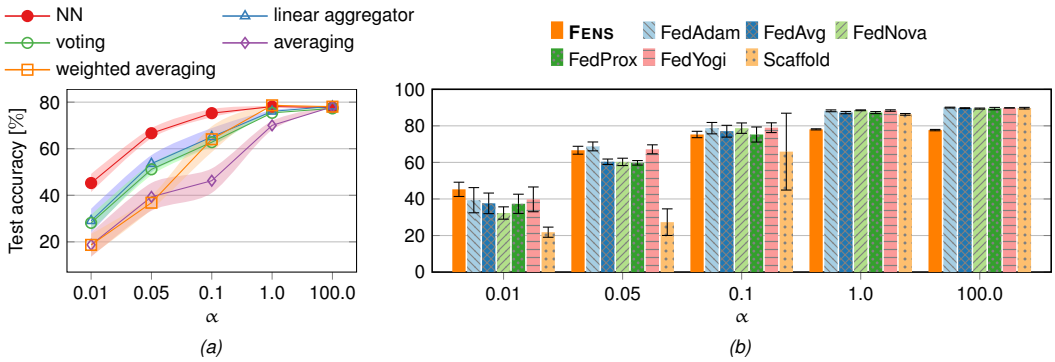

Figure 2: **(a) FENS aggregation methods:** NN performs the best. **(b) FENS vs FL:** FENS achieves competitive accuracies when the heterogeneity is very high ($\alpha \leq 0.1$). However, FL is significantly better when the data tends to be independent and identically distributed (IID) ($\alpha \geq 1.0$). Numerical values for all plots are included in Tables 8 and 9 (Appendix D) for reference.

and train for 100 communication rounds (details in Appendix B.2). The server assumes full client participation, *i.e.*, all 20 clients participate in each round, which is common in cross-silo FL with a limited number of clients. Each client performs 2 local epochs per round. While we run 100 communication rounds, we found all algorithms to converge much before the 100^{th} round.

One-shot FL baselines. We compare FENS against three one-shot baselines: the auxiliary dataset based FEDKD (Gong et al., 2022), data-free FEDCVAE-ENS (Heinbaugh et al., 2023) and the standard one-round FEDAVG. For one-shot FEDKD, we use the setup described by Gong et al. (2022) without inference quantization, using CIFAR-100 as the public dataset for distillation. For FEDCVAE-ENS, we use the same CVAE as well as the CNN classifier from Heinbaugh et al. (2023). Moreover, we experiment with ResNet-8 as the classifier. We run all experiments using the best-reported hyperparameters for the CIFAR-10 task in Heinbaugh et al. (2023). In addition to these one-shot baselines, we implement FEDAVG with gradient compression, following the sparsification and quantization schemes of STC (Mora et al., 2022). In particular, we set the quantization precision and sparsity level to match the communication cost of FENS (details in Appendix C.1) and the remaining setup to the same as FL.

4.1.3 RESULTS

Each experiment 5 ran times with different seeds and we report results with a 95% confidence interval.

FENS aggregation methods. Figure 2(a) depicts the accuracy of all aggregation methods for 5 values of α, showing a predictable increase as heterogeneity decreases. Remarkably, all methods converge to a single accuracy of $\sim$78% at $\alpha = 100$. The two trainable methods of voting and linear aggregation perform similarly. However, NN significantly outperforms all other methods when heterogeneity is high, suggesting the importance of non-linearity for effective aggregation. Although both plain and weighted averaging underperform at high heterogeneity, the latter quickly improves as heterogeneity decreases. Given these results, we pick the NN approach for comparison against the FL methods.

FENS vs FL. Figure 2(b) charts the test accuracy of FENS against the FL algorithms across the heterogeneity spectrum. Under extreme heterogeneity ($\alpha = 0.01$), FENS surpasses all FL algorithms, and by around 6% in accuracy for the best performing FEDYOGI. Similarly, at very high heterogeneity ($\alpha = 0.05$), FENS remains competitive to FEDYOGI and FEDADAM while surpassing all the others. The accuracy values at $\alpha = 0.1$ are also similar. Beyond this point, FL demonstrates better performance than FENS. In nearly IID data settings ($\alpha \geq 1.0$), FL algorithms achieve 12% higher accuracy than FENS. This happens because iterative FL thrives when learning from numerous clients with similar data distributions. However, when the local data distributions differ, the global model struggles to learn shared characteristics. In such scenarios, FENS can effectively harness diverse local classifiers, thanks to the NN aggregator. We investigate this phenomenon further in Section 4.2.

We analyze communication costs in Table 2 for $\alpha = \{0.1, 1\}$ by computing the number of rounds required to reach a pre-defined target accuracy. We set the target accuracy as the one achieved by

FENS. Although FL attains superior final accuracy at $\alpha = 1$, we still compare the corresponding differences in communication cost for scenarios where a lower target might be acceptable. FENS notably reduces total communication costs compared to FL, despite increased inference costs. Overall, FENS can save $6.4 - 9.1\times$ communication when the data is heterogeneous and $2.5 - 3.5\times$ under IID data.

Table 2: FENS vs FL communication costs comparison on the CIFAR-10 dataset. Training, inference, and total costs are presented in gibibytes (GiB).

Algorithm	$\alpha = 0.1$ (non-IID), Target = 75.2%					$\alpha = 1$ (IID), Target = 77.6%				
	Rounds	Train	Inference	Total	Factor	Rounds	Train	Inference	Total	Factor
FEDAVG	85	62.21	0.37	62.58	**8.5**×	34	24.88	0.37	25.25	**3.4**×
FEDPROX	91	66.60	0.37	66.97	**9.1**×	35	25.61	0.37	25.98	**3.5**×
FEDYOGI	64	46.84	0.37	47.21	**6.4**×	27	19.76	0.37	20.13	**2.7**×
FEDADAM	75	54.89	0.37	55.26	**7.5**×	25	18.29	0.37	18.66	**2.5**×
SCAFFOLD	92	67.33	0.37	67.70	**9.2**×	35	25.61	0.37	25.98	**3.5**×
FEDNOVA	64	46.84	0.37	47.21	**6.4**×	25	18.29	0.37	18.66	**2.5**×
FENS	**1**	**0.37**	**6.95**	**7.32**	–	**1**	**0.37**	**6.95**	**7.32**	–

FENS vs one-shot FL. Table 3 provides a comprehensive evaluation of test accuracy of FENS against one-shot FL baselines and gradient compression in FL. We observe that all one-shot methods struggle under high heterogeneity where FENS consistently outperforms the baselines. FEDCVAE-ENS exhibits consistent accuracy across different heterogeneity levels. However, it remains sensitive to the choice of classifier architecture and struggles to perform well on CIFAR-10, despite its strong performance on simpler tasks like MNIST (Heinbaugh et al., 2023). In homogeneous data distributions, both one-shot FEDKD and FENS exhibit similar performance. However, in more heterogeneous settings, FENS outperforms one-shot FEDKD by a substantial margin, up to 26% higher accuracy at $\alpha = 0.05$. Furthermore, FENS consistently delivers superior results compared to gradient compression when operating under similar communication budgets.

Table 3: FENS vs one-shot FL algorithms on CIFAR-10.

Algorithm	Test Accuracy [%]				
	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 100$
FEDAVG one-round	10.52±1.03	16.78±4.88	10.34±1.16	15.74±1.99	17.49±1.72
FEDCVAE-ENS (ResNet-8)	21.54±3.92	25.83±1.81	29.07±1.61	27.86±1.38	26.62±1.16
FEDCVAE-ENS (default CNN)	32.85±1.23	32.10±2.05	32.90±0.97	31.22±1.49	29.51±0.73
FEDKD	23.07±4.40	40.12±5.18	60.48±13.90	**78.43±1.21**	**79.41±0.27**
Grad. Compression	34.5±0.92	40.54±1.33	57.64±0.66	71.06±0.44	73.68±0.13
FENS-NN (ours)	**45.26±4.45**	**66.65±2.47**	**75.27±1.99**	78.08±0.35	77.67±0.42

4.2 UNDERSTANDING THE PERFORMANCE OF FENS

In this section, we aim to assess the performance of FENS under high data heterogeneity. As noted earlier, FL struggles to generate a good global model when the local datasets of clients significantly differ. In such cases, an ensemble of models, each specialized on data belonging to distinct clients, shows better performance as demonstrated by our results on CIFAR-10 (Section 4.1). The quality of ensembles is highly dependent on the quality of local models, which in turn depends on the amount of local data held by the clients. As local models improve at generalizing locally, the overall performance of the ensemble is enhanced. On the other hand, more volume of data does not analogously benefit FL due to high data heterogeneity. We validate this intuition through the following experiments on the SVHN dataset.

4.2.1 SETUP & HYPERPARAMETERS

We study the performance of FL and FENS by progressively increasing the volume of data held by the clients. To this end, we consider the classification task on the SVHN dataset due to the availability

of an extended training set of 604 388 samples, *i.e.*, more than $10\times$ bigger than CIFAR-10. We then experiment with fractions ranging from 10 to 100% of the total training set. For each subset, we split it into 90-10% to obtain $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{proxy}}$, respectively. We then compare FENS with FEDAVG (FL baseline) on two levels of heterogeneity: $\alpha = 0.1$ (non-IID) and $\alpha = 1.0$ (IID). We tune the learning rate for FEDAVG (details in Appendix B.3) and keep the remaining setup as in previous experiments.

4.2.2 RESULTS

Figure 3 shows the results and confirms our prior insight behind the effective performance of FENS. Specifically, we observe that the growing volume of training data benefits FENS much more than FEDAVG. When the data distribution is homogeneous ($\alpha = 1$), the performance of FENS improves faster than FEDAVG, but still remains behind. On the other hand, under high heterogeneity ($\alpha = 0.1$), FENS quickly catches up with the performance of FEDAVG, matching the same accuracy when using the full training set. In summary, our observations can be attributed to the deterioration in FL performance, coupled with the efficacy of ensembles in harnessing heterogeneous models in the face of significant data heterogeneity.

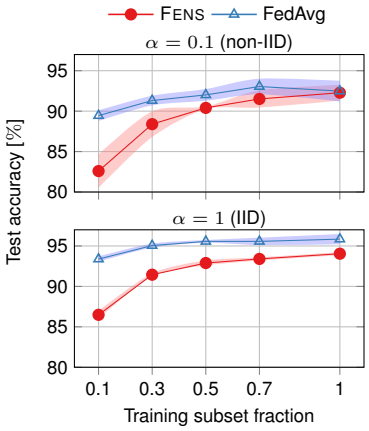

Figure 3: Performance of FENS with increasing number of samples.

4.3 FLAMBY MEDICAL BENCHMARK

In this section, we evaluate the performance of FENS on the real-world cross-silo FLamby benchmark (Ogier du Terrail et al., 2022).

4.3.1 DATASETS AND TASKS

The FLamby suite consists of seven datasets with a diverse set of tasks, including 3D segmentation, survival analysis, binary and multi-class classification. Being a cross-silo FL benchmark, the number of clients in each dataset is relatively low and varies between two and six. Owing to the large size of medical data (scans, slides *etc.*), datasets are quite large (up to 850 GiB) and take significantly long preprocessing and training times. We focus on the classification tasks and experiment on 3 datasets: Fed-Camelyon16, Fed-Heart-Disease, and Fed-ISIC2019. Table 4 (Appendix A) presents an overview of the selected tasks. The datasets consist of a natural non-IID partitioning across clients. We reserve 10%, 15%, and 15% of the client datasets as $\mathcal{D}_{\text{proxy}}$ at the server for Fed-Heart-Disease, Fed-Camelyon16, and Fed-ISIC2019, respectively. In other words, FENS clients train on the portion of local data not in $\mathcal{D}_{\text{proxy}}$ while the server trains the NN aggregator using $\mathcal{D}_{\text{proxy}}$. We note that clients in other FL algorithms use 100% of their local dataset for training. This ensures a fair comparison such that all methods observe the same amount of training data. Finally, all performances are reported on the original testing split $\mathcal{D}_{\text{test}}$.

4.3.2 SETUP AND HYPERPARAMETERS

In FLamby, the authors benchmark various FL algorithms (*strategies*), including FEDAVG, FEDPROX, FEDYOGI, and SCAFFOLD, and compare them to the *pooled training* and client local baselines. While these benchmarks were obtained after extensive tuning of hyperparameters, the authors purposefully restricted the number of rounds to be approximately the same as the number of epochs required to train on pooled data (see Ogier du Terrail et al. (2022)). Since this might not reflect true FL performance, we rerun all FL strategies to convergence using the reported tuned parameters. Precisely, we run up to $10\times$ more communication rounds than evaluated in the FLamby benchmark. For FENS, each client performs local training with the same hyperparameters as the client local baselines in FLamby while the server trains the NN aggregator using a similar configuration as before. For the one-shot FEDAVG and FEDPROX baselines, we additionally tune the number of local updates. We remark that FEDKD is infeasible in these settings since it requires a public dataset for distillation, unavailable in the medical setting. FEDCVAE-ENS is also infeasible due to the difficulty in learning good decoders for medical input data, a conclusion supported by its poor performance on the comparatively simpler CIFAR-10 task (Table 3).

4.3.3 RESULTS

Figure 4 shows the results with the first row comparing FENS against iterative FL algorithms and the second row against one-shot FL and the client local baselines. When comparing to iterative FL, we observe that FENS is on par for the Fed-Heart-Disease dataset and performs better for Fed-Camelyon16. However, the iterative FL algorithms achieve significantly higher accuracy on the Fed-ISIC2019 dataset. We justify this poor performance of FENS with some key observations on the client local baselines.

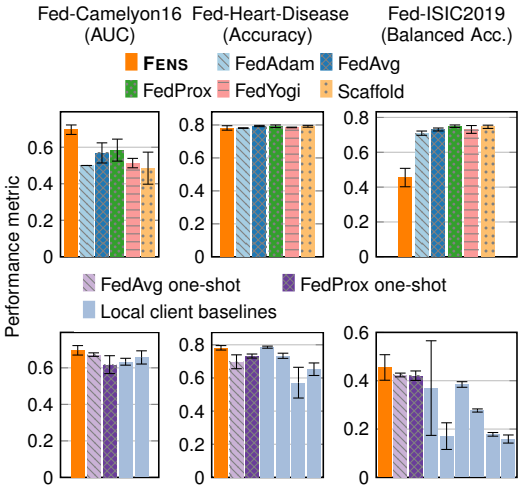

Figure 4: **FENS in FLamby.** FENS is on par with iterative FL (row-1), except when local models are weak (Fed-ISIC2019) while remaining superior in the one-shot setting (row-2). Numerical results included in Tables 10 to 15 (Appendix D).

For both Fed-Camelyon16 and Fed-Heart-Disease datasets, the client local baselines perform well, while FL performance is affected by high data heterogeneity. The deterioration is more significant for Fed-Camelyon16, which learns on large brain slides (10000×2048), than for Fed-Heart-Disease, which learns on tabular data. In such scenarios, FENS can harness diverse local classifiers to attain good performance. In contrast, in the case of Fed-ISIC2019, the clients possess highly unbalanced quantities of data samples (Table 4, Appendix A), producing weak local classifiers. Thus, FENS struggles to keep up with the performance of iterative FL, matching our intuition from Section 4.2. However, we note that FENS achieves superior performance over one-shot FEDAVG and one-shot FEDPROX, while performing at least as well as the best client local baseline. Overall, these observations for FL algorithms have spurred new interest in developing a better understanding of performance on heterogeneous cross-silo datasets (Ogier du Terrail et al., 2022). We show that FENS remains a strong competitor in such settings.

5 DISCUSSION AND CONCLUSION

Limitations. One limitation is the privacy aspect, as client models are shared and not aggregated. This leaves room for attacks like membership inference (Melis et al., 2019). Future work can explore mitigation techniques such as differential privacy (Geyer et al., 2017) or trusted execution environments (Messaoud et al., 2022) to address this concern. Another limitation is resource usage at inference time, as FENS requires additional memory and compute resources to store and process the complete global model. However, this is more manageable in cross-silo federated setups with reliable and powerful clients.

Benefits. In addition to the benefits of low communication costs and rapid training, FENS provides three important advantages. Firstly, it supports model heterogeneity, allowing for different model architectures across federated clients, which is beneficial when organizations or entities have specific requirements or constraints for their models (Li & Wang, 2019). Secondly, FENS aligns with the design principles of the popular SISA framework (Bourtoule et al., 2021) for practical machine unlearning, towards the goal of the *right to be forgotten* in GDPR (Mantelero, 2013). This enables efficient unlearning compared to traditional FL where unincorporating knowledge from a single global model can be very costly. Lastly, FENS enables model reusability by allowing clients to contribute previously trained models Li et al. (2021), fostering collaboration without the need to start training from scratch. This approach prevents resource wastage and lowers entry barriers for clients.

To conclude, this paper investigates Federated Ensembles (FENS), with a specific focus on cross-silo federated settings. FENS emphasizes local training and one-shot model sharing, thereby reducing communication costs and accelerating training. The experiments on a wide range of tasks demonstrated that FENS with the NN aggregator is notably effective in settings with high data heterogeneity, reducing the communication costs by a factor of up to $9.1\times$. Furthermore, FENS offers additional benefits of model heterogeneity, machine unlearning, and model reusability.

REPRODUCIBILITY STATEMENT

We have undertaken several steps to ensure the integrity, reproducibility and replicability of FENS. We have provided an extensive description of FENS in the Section 3, describing formally its algorithmic procedure in Section 3.2 and Section 3.3. To facilitate the reproducibility of the experimental results, the complete source code used for the evaluation of FENS will be made publicly available and a link will be added in the final paper version. We have used publicly available ML models and datasets. Each presented evaluation carries a subsection on setup and hyperparameters (Sections 4.1.2, 4.2.1 and 4.3.2) with sufficient information to reproduce all our results. We believe that these efforts will aid researchers in understanding, replicating, and building upon FENS.

ETHICS STATEMENT

We affirm that the medical datasets employed in our experiments within the FLamby benchmark were acquired and utilized in strict accordance with their respective licensing agreements and ethical guidelines. We obtained the necessary permissions and approvals from the appropriate authorities and/or institutions responsible for data collection, and we adhered to all relevant ethical standards and regulations.

REFERENCES

Ruth Ben-Yashar and Jacob Paroush. Optimal decision rules for fixed-size committees in polychotomous choice situations. *Social Choice and Welfare*, 18(4):737–746, 2001.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. In *MLSys*, 2019. URL https://mlsys.org/Conferences/2019/doc/2019/193.pdf.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159. IEEE, 2021.

Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, 2020. doi: 10.1109/IJCNN48605.2020.9207469.

Hong-You Chen and Wei-Lun Chao. Fed{be}: Making bayesian model ensemble applicable to federated learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=dgtpE6gKjHn.

Mudasir A Ganaie, Minghui Hu, AK Malik, M Tanveer, and PN Suganthan. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151, 2022.

Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

Xuan Gong, Abhishek Sharma, Srikrishna Karanam, Ziyan Wu, Terrence Chen, David Doermann, and Arun Innanje. Preserving privacy in federated learning with ensemble cross-domain knowledge distillation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):11891–11899, Jun. 2022. doi: 10.1609/aaai.v36i11.21446. URL https://ojs.aaai.org/index.php/AAAI/article/view/21446.

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

Jenny Hamer, Mehryar Mohri, and Ananda Theertha Suresh. Fedboost: A communication-efficient algorithm for federated learning. In *International Conference on Machine Learning*, pp. 3973–3983. PMLR, 2020.

Florian Hartmann, Sunah Suh, Arkadiusz Komarzewski, Tim D Smith, and Ilana Segall. Federated learning for ranking browser history suggestions. *arXiv preprint arXiv:1911.11807*, 2019.

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*, pp. 770–778, 2016.

Clare Elizabeth Heinbaugh, Emilio Luz-Ricca, and Huajie Shao. Data-free one-shot federated learning under very high statistical heterogeneity. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=_hb4vM3jspB.

Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2127–2136. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/ilse18a.html.

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.

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*, pp. 5132–5143. PMLR, 2020.

Michael Kearns. Thoughts on hypothesis boosting. *Unpublished manuscript*, 45:105, 1988.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. 55(5), 2014. URL https://www.cs.toronto.edu/~kriz/cifar.html.

Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *NeurIPS Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019.

Qinbin Li, Bingsheng He, and Dawn Song. Practical one-shot federated learning for cross-silo setting. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 1484–1490. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/205. URL https://doi.org/10.24963/ijcai.2021/205. Main Track.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Wenqi Li, Fausto Milletarì, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M Jorge Cardoso, et al. Privacy-preserving federated brain tumour segmentation. In *Machine Learning in Medical Imaging: 10th International Workshop, MLMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 10*, pp. 133–141. Springer, 2019.

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.

Jiachen Liu, Fan Lai, Yinwei Dai, Aditya Akella, Harsha Madhyastha, and Mosharaf Chowdhury. Auxo: Heterogeneity-mitigating federated learning via scalable client clustering. *arXiv preprint arXiv:2210.16656*, 2022.

Alessandro Mantelero. The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235, 2013.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pp. 691–706. IEEE, 2019.

Aghiles Ait Messaoud, Sonia Ben Mokhtar, Vlad Nitu, and Valerio Schiavoni. Shielding federated learning systems against inference attacks with arm trustzone. In *Proceedings of the 23rd ACM/IFIP International Middleware Conference*, pp. 335–348, 2022.

Alessio Mora, Luca Foschini, and Paolo Bellavista. Structured sparse ternary compression for convolutional layers in federated learning. In *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pp. 1–5, 2022. doi: 10.1109/VTC2022-Spring54318.2022.9860833.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, Othmane Marfoq, Erum Mushtaq, Boris Muzellec, Constantin Philippenko, Santiago Silva, Maria Teleńczuk, Shadi Albarqouni, Salman Avestimehr, Aurélien Bellet, Aymeric Dieuleveut, Martin Jaggi, Sai Praneeth Karimireddy, Marco Lorenzi, Giovanni Neglia, Marc Tommasi, and Mathieu Andreux. Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 5315–5334. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/232eee8ef411a0a316efa298d7be3c2b-Paper-Datasets_and_Benchmarks.pdf.

Robi Polikar. Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6 (3):21–45, 2006.

Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LkFG3lB13U5.

MyungJae Shin, Chihoon Hwang, Joongheon Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Xor mixup: Privacy-preserving data augmentation for one-shot federated learning. *arXiv preprint arXiv:2006.05148*, 2020.

Nir Shlezinger, Erez Farhan, Hai Morgenstern, and Yonina C. Eldar. Collaborative inference via ensembles on the edge. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8478–8482, 2021. doi: 10.1109/ICASSP39728.2021.9414740.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Minxue Tang, Xuefei Ning, Yitu Wang, Jingwei Sun, Yu Wang, Hai Li, and Yiran Chen. Fedcor: Correlation-based active client selection strategy for heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10102–10111, 2022.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=BkluqlSFDS.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020b.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pp. 7252–7261. PMLR, 2019.

Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Jianghe Xu, Shouhong Ding, and Chao Wu. A practical data-free approach to one-shot federated learning with heterogeneity. *arXiv preprint arXiv:2112.12371*, 2021.

Yanlin Zhou, George Pu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. Distilled one-shot federated learning. *arXiv preprint arXiv:2009.07999*, 2020.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning*, pp. 12878–12889. PMLR, 2021.

A DATASETS

As mentioned in Section 4.3.1, we focus on classification tasks and experiment with 3 datasets in FLamby benchmark including Fed-Camelyon16, Fed-Heart-Disease and Fed-ISIC2019. Table 4 overviews the selected tasks in this work.

Table 4: Overview of selected datasets and tasks in FLamby. We defer additional details to (Ogier du Terrail et al., 2022).

DATASET	INPUT (x)	PREDICTION (y)	TASK TYPE	# CLIENTS	# EXAMPLES PER CLIENT	MODEL	METRIC
Fed-Camelyon16	Slides	Tumor on Slide	Binary Classification	2	239, 150	DeepMIL Ilse et al. (2018)	AUC
Fed-Heart-Disease	Patient Info.	Heart Disease	Binary Classification	4	303, 261, 46, 130	Logistic Regression	Accuracy
Fed-ISIC2019	Dermoscopy	Melanoma Class	Multi-class Classification	6	12413, 3954, 3363, 225, 819, 439	EfficientNet Tan & Le (2019) + Linear layer	Balanced Accuracy

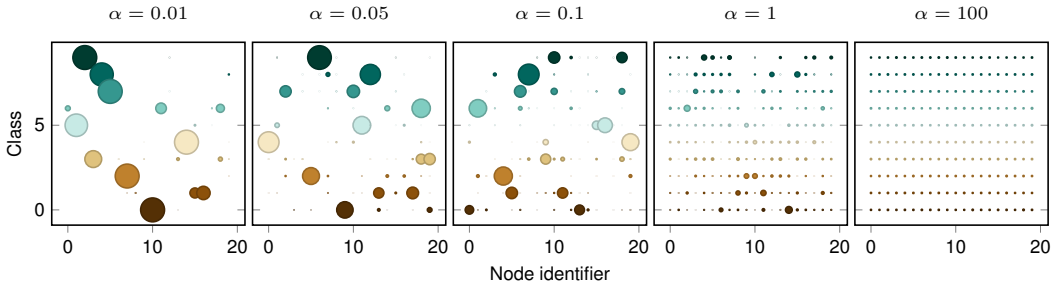

Figure 5: Visualizing the effect of changing α on the CIFAR-10 dataset. Dot size corresponds to number of samples of a given class in a given node.

B ADDITIONAL EXPERIMENTAL DETAILS

B.1 AGGREGATORS EVALUATED IN FENS

In Section 3.3, we evaluate the methods of averaging, weighted averaging, polychotomous voting, linear aggregator and NN aggregator for FENS. For training the linear aggregator, we follow the exact setup of Hamer *et al*. (Hamer et al., 2020), which uses the mirror descent algorithm with Kullback-Leibler (KL) divergence as the Bregman divergence. The linear aggregator trains for 2000 epochs with a learning rate of 10^{-4}. Recall from Section 3.3 that polychotomous voting requires the competency matrix of each expert (*i.e*., each client). The server evaluates this competency matrix for each client's model on the $\mathcal{D}_{\text{proxy}}$ dataset which can be seen as a form of training. The linear aggregation and NN aggregation schemes also utilize $\mathcal{D}_{\text{proxy}}$ for training at the federated server.

B.2 HYPERPARAMETER TUNING FOR CIFAR-10

Below we describe our tuning procedure derived from several previous works Ogier du Terrail et al. (2022); Li et al. (2020); Reddi et al. (2021). For the FEDAVG algorithm, we tune the client learning rate (η_l) over the values $\{0.1, 0.01, 0.001, 0.0001\}$ separately for every $\alpha \in \{0.01, 0.05, 0.1, 1.0, 100\}$. For the FEDPROX algorithm, our grid space was $\{0.1, 0.01\}$ and $\{1, 0.1, 0.01\}$ for the client learning rate (η_l) and the proximal parameter (μ) respectively. This was again separately tuned for every value of $\alpha \in \{0.01, 0.05, 0.1, 1.0, 100\}$. For the FEDYOGI and the FEDADAM algorithm, we consider the grid space of $\{0.1, 0.01, 0.001, 0.0001\}$ and $\{10, 1, 0.1, 0.01, 0.001\}$ for the client learning rate (η_l) and the server learning rate (η_s) respectively. This explodes the search significantly when tuning for every value of α. From our tuning results for the FEDAVG and the FEDPROX algorithm, we noticed that the optimal parameter values were the same within the following two subgroups of α – $\{0.01, 0.05\}$ and $\{0.1, 1, 100\}$ (Table 5). Hence, to keep the tuning tractable, we tune only for one α

in each subgroup and reuse the values for other alphas within the same subgroup. For the FEDNOVA algorithm, we use the version with both client and global momentum which was reported to perform the best (Wang et al., 2020b). We consider the search space $\{0.005, 0.01, 0.02, 0.05, 0.08\}$ for the client learning rate (η_l) as done by the authors Wang et al. (2020b) and tune separately for every value of α. Finally, for the SCAFFOLD algorithm, we consider the search spaces $\{0.1, 0.01, 0.001, 0.0001\}$ and $\{1.0, 0.1, 0.01\}$ for η_l and η_s respectively and also tune separately for every α. Table 5 details the hyperparameters obtained after tuning.

Table 5: Best hyperparameters obtained for the different algorithms on the CIFAR-10 dataset.

	FEDAVG	FEDPROX		FEDYOGI		FEDADAM		FEDNOVA	SCAFFOLD	
α	η_l	η_l	μ	η_l	η_s	η_l	η_s	η_l	η_l	η_s
0.01	0.01	0.01	0.01	0.01	0.01	-	-	0.0001	0.0001	1.0
0.05	0.01	0.01	0.01	-	-	0.01	0.01	0.02	0.01	0.1
0.1	0.1	0.1	0.01	-	-	-	-	0.005	0.01	1.0
1.0	0.1	0.1	0.01	-	-	-	-	0.005	0.01	1.0
100	0.1	0.1	0.01	0.01	0.01	0.01	0.01	0.005	0.1	1.0

B.3 HYPERPARAMETER TUNING FOR SVHN

Table 6: Best hyperparameters obtained for FEDAVG on the SVHN dataset.

	FEDAVG
α	η_l
0.01	0.1
0.05	0.1
0.1	0.1
1.0	0.1
100	0.1

For the FEDAVG algorithm, we tune the client learning rate (η_l) over the search space $\{0.1, 0.01, 0.001, 0.0001\}$ separately for every $\alpha \in \{0.01, 0.05, 0.1, 1.0, 100\}$. Table 6 reports the best obtained learning rates.

C ADDITIONAL IMPLEMENTATION DETAILS

C.1 FEDAVG WITH GRADIENT COMPRESSION

To implement FEDAVG with gradient compression, we followed the sparsification and quantization schemes of STC (Mora et al., 2022). In the test bed, we experimented with different precision levels for quantization. After careful evaluation, we determined that 8-bit quantization was the minimum precision level that achieved reasonable accuracy. Hence, we chose 8-bit quantization instead of the ternary quantization method described in STC. We use the sparsity level of 40% to compete with the communication cost of FENS. Finally, we keep the remaining setup to be the same as FL. The communication costs of the considered algorithms including FENS are reported in Table 7. We note that in the case of FEDKD, the CIFAR-100 public dataset must be downloaded by each client. Consequently, we present the associated cost of downloading the dataset ($20 \times 162\,\mathrm{MiB}$) in parentheses.

C.2 POLYCHOTOMOUS VOTING

Polychotomous voting Ben-Yashar & Paroush (2001), was originally developed in social choice theory to reach a collective decision when offered C alternatives (classification labels in our case) in a committee of M experts (clients in our case). This method requires as input: *(i)* classwise "competency" scores of each client: $P_i^c(r)$ indicating the probability of i^{th}-client to vote for label c

Table 7: Communication costs associated with one-shot methods and gradient compression algorithm.

Algorithm	Communication Cost [GiB]		
	Training	Inference	Total
FEDKD	0.04 (+ 3.16)	0.37	0.41 (+ 3.16)
Grad. Compression	7.35	0.04	7.39
FENS	0.37	6.98	7.35

when the ground truth is r; *(ii)* $p_{prior}(r)$: prior probability distribution over correct alternatives r; and *(iii)* the "benefit" vector of the committee: $B(c|r)$ indicating the committee's benefit in choosing label c when the correct class is r. Given this information, Ben-Yashar and Paroush Ben-Yashar & Paroush (2001) derive a criterion for the optimal decision that maximizes expected utility. This criterion is not computed using a closed-form expression, and we generically express it as

$$f(\pi_1, \ldots, \pi_M; \boldsymbol{P}_1, \ldots, \boldsymbol{P}_M; \boldsymbol{p}_{prior}; B) \tag{6}$$

where f corresponds to a procedure that evaluates and compares benefits for each choice $c \in [C]$ given the competency matrices $\{\boldsymbol{P}_i\}_{i=1}^M$, the priors $\boldsymbol{p}_{prior}$ and the benefit function B. Since the competency matrices for each client model are not directly available in our distributed setting, we learn them by evaluating the models on $\mathcal{D}_{proxy}$. Hence, polychotomous voting qualifies as a trainable method. We further use a simple benefit function for our experiments that assigns $B(c/r) = 1$ when $c = r$ (correct choice) and $B(c/r) = 0$ when $c \neq r$ (incorrect choice) and set the prior to be uniform over the set of labels.

D NUMERICAL RESULTS

In this section, we include the numerical values in Tables 8 to 15 corresponding to the plots presented in Sections 4.1 and 4.3 for a complete reference.

Table 8: FENS aggregation methods. Results of Figure 2(a).

Algorithm	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 100$
Averaging	18.73±5.90	39.33±6.81	46.25±6.00	70.03±2.22	78.00±0.19
Weighted Averaging	18.69±5.43	36.80±3.89	64.06±6.32	**78.63**±0.41	78.05±0.19
Polychotomous Voting	28.15±3.87	51.14±2.86	62.80±1.99	75.40±0.43	77.31±0.27
Linear Aggregator	29.04±5.88	53.58±4.16	65.07±4.31	76.04±0.48	**78.21**±0.18
NN Aggregator	**45.26**±4.45	**66.65**±2.47	**75.27**±1.99	78.08±0.35	77.67±0.42

Table 9: FENS vs SOTA FL algorithms on CIFAR-10. Results of Figure 2(b).

Algorithm	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 100$
FEDADAM	39.324 ± 7.855	**68.748** ± 2.762	78.736 ± 3.552	88.228 ± 0.565	**89.980** ± 0.271
FEDAVG	37.600 ± 6.428	60.344 ± 1.705	77.062 ± 3.678	87.256 ± 0.711	89.690 ± 0.275
FEDNOVA	32.316 ± 3.844	60.280 ± 2.300	78.732 ± 3.252	**88.508** ± 0.259	89.432 ± 0.339
FEDPROX	37.344 ± 6.001	59.772 ± 1.413	75.250 ± 4.705	87.260 ± 0.631	89.446 ± 0.731
FEDYOGI	39.788 ± 7.726	67.168 ± 2.793	**78.980** ± 3.047	88.362 ± 0.467	89.818 ± 0.068
SCAFFOLD	21.824 ± 3.168	27.308 ± 8.303	65.892 ± 23.980	86.284 ± 0.525	89.660 ± 0.502
FENS	**45.264** ± 4.452	66.652 ± 2.472	75.270 ± 1.989	78.082 ± 0.352	77.670 ± 0.422

Table 10: Figure 4 results. FENS vs. iterative FL – Fed-Camelyon16 (row 1).

Algorithm	AUC
FEDADAM	0.500±0.000
FEDAVG	0.569±0.063
FEDPROX	0.584±0.069
FEDYOGI	0.513±0.026
SCAFFOLD	0.485±0.100
FENS	**0.696**±0.030
Pooled Training	0.728 ± 0.041

Table 11: Figure 4 results. FENS vs. one-shot FL – Fed-Camelyon16 (row 2).

Algorithm	AUC
Client 0	0.633 ± 0.021
Client 1	0.657 ± 0.041
FEDAVG-OS	0.673 ± 0.007
FEDPROX-OS	0.618 ± 0.043
FENS	**0.696** ± 0.030
FEDAVG	0.569 ± 0.063
FEDPROX	0.584 ± 0.069

Table 12: Figure 4 results. FENS vs. iterative FL – Fed-Heart-Disease (row 1).

Algorithm	Accuracy
FEDADAM	0.781±0.002
FEDAVG	**0.794**±0.004
FEDPROX	0.792±0.009
FEDYOGI	0.785±0.002
SCAFFOLD	0.792±0.006
FENS	0.782±0.015
Pooled Training	0.795±0.003

Table 13: Figure 4 results. FENS vs. one-shot FL – Fed-Heart-Disease (row 2).

Algorithm	Accuracy
Client 0	**0.786**±0.007
Client 1	0.734±0.018
Client 2	0.572±0.106
Client 3	0.653±0.043
FEDAVG-OS	0.698±0.048
FEDPROX-OS	0.732±0.014
FENS	0.782±0.015
FEDAVG	0.794±0.004
FEDPROX	0.792±0.009

Table 14: Figure 4 results. FENS vs. iterative FL – Fed-ISIC2019 (row 1).

Algorithm	Balanced Accuracy
FEDADAM	0.710±0.011
FEDAVG	0.731±0.007
FEDPROX	0.750±0.006
FEDYOGI	0.731±0.020
SCAFFOLD	**0.746**±0.008
FENS	0.455±0.060
Pooled Training	0.762±0.010

Table 15: Figure 4 results. FENS vs. one-shot FL – Fed-ISIC2019 (row 2).

Algorithm	Balanced Accuracy
Client 0	0.370±0.223
Client 1	0.171±0.063
Client 2	0.384±0.013
Client 3	0.277±0.007
Client 4	0.178±0.009
Client 5	0.159±0.019
FEDAVG-OS	0.424±0.007
FEDPROX-OS	0.421±0.017
FENS	**0.455**±0.060
FEDAVG	0.731±0.007
FEDPROX	0.750±0.006