

Article

Not peer-reviewed version

Federated Distillation Methodology for Label-based Group Structures

[Geon Hee Yang](#)^{*} and [Hyunchul Tae](#)^{*}

Posted Date: 12 October 2023

doi: 10.20944/preprints202310.0791.v1

Keywords: federated learning; distillation; federated distillation; clustering





Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Federated Distillation Methodology for Label-Based Group Structures

Geonhee Yang  and Hyunchul Tae * 

Digital Healthcare Research Department, Korea Institute of Industrial Technology, Cheonan 31056, Republic of Korea

* Correspondence: sage@kitech.re.kr

Abstract: In federated learning (FL), clients train models locally without sharing raw data, ensuring data privacy. In particular, federated distillation transfers knowledge to clients regardless of the model architecture. However, when groups of clients with different label distributions exist, sharing the same knowledge among all clients becomes impractical. To address this issue, this paper presents an approach that clusters clients based on the output of a client model trained using their own data. The clients are clustered based on the predictions of their models for each label on a public dataset. Evaluations on MNIST and CIFAR show that our method effectively finds group identities, increasing accuracy by up to 75% over existing methods when the distribution of labels differs significantly between groups. In addition, we observed significant performance improvements on smaller client groups, bringing us closer to fair FL.

Keywords: federated learning; distillation; federated distillation; clustering

1. Introduction

Federated learning (FL) enables multiple clients to contribute to training a global machine learning model without sharing their data with a central server [1]. Clients perform computations on their local data and send only the model updates to a central server, which aggregates them to improve the global model. The global model is then redistributed to the clients for subsequent training rounds. This framework ensures data safety by storing data only on client devices, thereby minimizing the risk of breaches [2]. In the context of healthcare, this approach is particularly valuable because it enables collaborative research and model training across multiple medical institutions while complying with strict privacy regulations and minimizing the risk of exposing sensitive patient data [3].

Distillation is a machine learning technique that trains a simpler model, called a student, to mimic the actions of a more complex model, called a teacher, which typically improves efficiency without sacrificing accuracy. Federated distillation extends this approach to a decentralized setting, allowing many devices to train a student model collaboratively while keeping their data localized [4]. Recently, federated distillation has attracted considerable attention. Federated distillation captures and communicates the learning experience through logits, which are the pre-activation function outputs of individually trained models. This approach significantly reduces the communication overhead compared to traditional FL [4]. It also provides a balance between the flexibility and security. Clients can use models suitable for their computational capabilities [5]. At the same time, the risks of information exposure are significantly reduced by transmitting only distilled knowledge through logits rather than raw data, thereby increasing the data privacy level [6].

However, certain client groups may have unique labels defined by variables, such as geography, demographics, or gender. Traditional methods rely on a uniform global logit, which results in reduced accuracy, particularly when the data have a distinct group structure. To illustrate this, consider FL between hospitals specializing in different types of medical treatments. A hospital specializing in cancer will have a dataset containing only different types of cancer, while another hospital specializing in infectious diseases will have images labeled "infection." In this context, the use of a uniform global logit compromises the quality of the global model, making it biased and less accurate.

Although clustering techniques exist in FL, to the best of our knowledge, no method has integrated clustering with federated distillation. Furthermore, most clustering algorithms in FL use model parameters for clustering. However, federated distillation does not exchange models. Therefore, a new clustering criterion was required. We proposed a method that classifies client models based on the number of times they predict each label. Figure 1 illustrates our algorithm, which utilizes information about clusters for effective distillation. In practice, the number of groups is often unknown. The algorithm we propose addresses this issue by using hierarchical clustering, which eliminates the need for prior knowledge of the number of clusters.

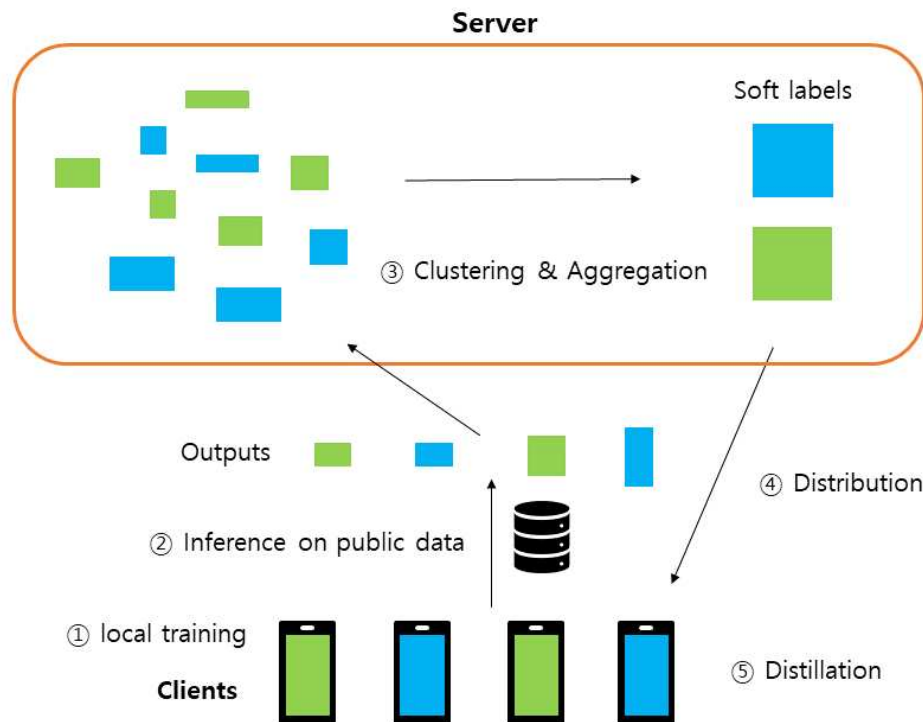


Figure 1. Federated distillation process using group structure. 1. Individual clients conduct local training using their own datasets. 2. Each client performs inference on a shared public dataset and forwards the results to a central server. 3. The server clusters the received outputs and obtains the soft label by averaging the outputs of the same group. 4. Clients receive the averaged soft labels corresponding to their groups. 5. Clients then perform distillation to align their model outputs with the received averaged soft labels.

In FL, fairness requires sensitive groups, such as gender and race, to not experience disparate outcome patterns such as different accuracy [1]. Unfortunately, minority social groups are often underrepresented in the training data. When the size of each client group varies, the existing methods significantly undermine the performance of minority client groups. On the other hand, our method performed well regardless of group size by assigning a logit that fits the distribution of the group data. This allowed us to get closer to a fair FL.

Guided by an empirical analysis of the esteemed MNIST and CIFAR datasets, we demonstrate that the clustering accuracy through prediction exceeds 90%. We also achieve high accuracy for each client model compared to traditional federated distillation methods in settings where an apparent group structure exists. Performance increased by up to 75%, and the greater the difference in data distribution between each group, the greater the advantage of our algorithm. We show that our algorithm is effective even when the data is sparse.

2. Related Works

2.1. Clustering in Federated Learning

In FL, several clustering criteria are available for categorizing and grouping clients based on various attributes. One example is Data Source-Based Clustering [7] organizes clients according to the origin of their data, such as X-rays or ultrasound, in medical settings. Geographical and Diagnosis-Based Clustering [8] groups clients based on their locations or shared diagnoses. Loss Function Values-Based Clustering [9,10] focuses on similar model behaviors as deduced from the loss function values. Clusters based on inherent data distribution seek to enhance model generalization by considering the intrinsic characteristics of the data. Model Parameter-Based Clustering [11,12] gathers clients with analogous model parameters, reflecting parallel learning stages. Gradient Information-Based Clustering [13,14] forms clusters by examining shared gradient information. Prototype-Based Clustering [15] simplifies the global model by forming clusters around generalized prototypes that represent distinct data patterns. These clustering criteria optimize FL to ensure efficient model training across diverse client datasets.

2.2. Federated Distillation

Federated distillation enables the adaptation of models to suit the computational capacity of a client [5] and minimizes information leakage during the sharing of high-dimensional model parameters [6]. FedMD enables heterogeneous FL among clients with different model structures by distilling knowledge from the server to the clients [5]. By contrast, Federated Group Knowledge Transfer (FedGKT) involves bidirectional distillation between clients and servers [16]. This method transfers the computational load from clients to servers, but raises privacy concerns. FedDF [17] uses unlabeled data to implement distillation while aggregating client-to-server logits across different model architectures. Distillation techniques have also been employed in One-Shot FL methods [18–20], which compress information from multiple client models into singular models.

3. Materials and Methods

3.1. Problem Definition

There are m clients. Each client k has a dataset $D_k := \{(x_i^k, y_i)\}_{i=1}^{N_k}$, where N_k represents the number of instances for client k . There are L client groups, each containing instances from a limited set of C_l classes, where $1 < C_l < C$ and C is the total number of classes. Clients also have access to an unlabeled public dataset $D_p := \{(x_i^p)\}_{i=1}^{N_p}$. Each client employs a model f_k with potentially different architectures.

3.2. Federated Distillation

In federated distillation, each client trains a local model and communicates its knowledge to the central server. We use a one-shot method [18] in which the client sends the trained results to the server once and receives the aggregated data in return. This approach minimizes the communication overhead and accelerates the learning process. The distillation process uses the standard KL divergence loss represented in Equation 1.

$$\text{KL}(p, q) = \sum_{c=1}^C p(c) \log \frac{p(c)}{q(c)} \quad (1)$$

where $p(c)$ and $q(c)$ denote the predicted probabilities of class c obtained from the client and group models, respectively. Mathematically, $p(c) = \sigma(f_k(x))$ and $q(c) = \sigma(\tilde{f}_l(x))$. $f_k(x)$ is the logit from the client's model, while $\tilde{f}_l(x)$ is the averaged logit from clients belonging to group l .

3.3. Clustered Federated Distillation

Our objective is to identify the group of each client and train a specialized model for each group using both D_p and D_k . We assume no prior knowledge of the groups, including the number of clusters L . The server aggregates the logits predicted by each client for D_p and then computes a count vector for each label predicted by each client k . This count vector is normalized, as described in Equations 2 and 3.

$$\mathbf{Count}_k = [\mathbf{Count}_k^c]_{c=1}^C, \quad \text{where} \quad \mathbf{Count}_k^c = \sum_{i=1}^{N_0} I(\arg\max(f_k(x_i^p)) = c) \quad (2)$$

In Equation 2, \mathbf{Count}_k represents a vector in which each element \mathbf{Count}_k^c denotes the number of instances in D_p classified into class c by client k 's model f_k . The function I serves as an indicator that returns 1 if the condition is true, and 0 otherwise.

$$\mathbf{NormCount}_k = \left[\frac{\mathbf{Count}_k^c - \min(\mathbf{Count}_k)}{\max(\mathbf{Count}_k) - \min(\mathbf{Count}_k)} \right]_{c=1}^C \quad (3)$$

We employed agglomerative clustering to identify the client groups. It is a hierarchical clustering method that starts with each data point as a separate group and iteratively merges the closest groups together. The `distance_threshold` serves as a key parameter for setting the maximum distance for merging groups. Equation 3 normalizes the count vectors to a $[0, 1]$ range to ensure the consistent application of `distance_threshold`.

Algorithm 1 outlines our Clustered Federated Distillation Learning method. Each client trains its model on a private dataset and predicts classes on a public dataset D_p . These predictions are sent to a centralized server that clusters the clients based on them. The server calculates the average logit $\tilde{f}_l(x^p)$ for each cluster and sends it back to the corresponding client. The clients then distill their models using the KL divergence loss as Eq 1, effectively addressing non-IID data distribution and enhancing overall model performance.

Algorithm 1 Clustered Federated Distillation framework

Input: Public dataset D_p , private dataset D_k , model of client k : f_k , $k = 1, \dots, m$, L group and l_c clients at group l .
Output: Trained model f_k .
Train: Each client trains f_k on D_k .
Predict: Each client predicts class $f_k(x^p)$ on D_p , and transmits the result to a central server.
Cluster: The server clusters using each client's prediction. Using Eq. 2, 3.
Aggregate: The server averages the logit for each cluster. $\tilde{f}_l(x^p) = \frac{1}{l_c} \sum_{k \in \text{Group } l} f_k(x^p)$.
Distribute: Each client receives own group's logit $\tilde{f}_l(x^p)$.
Distill: Each client model learns by distilling knowledge from $\tilde{f}_l(x^p)$. Using Eq. 1.

4. Results

4.1. Setting

To experiment with different group structures, we varied the number of classes per group, ranging from 2 to 5, and the number of groups ranging from 2, 4, 6, 8, and 10. We used the MNIST [21] dataset, which has a total of 10 classes. This implied that a single class often belonged to multiple groups. For the neural network architecture, we employed a simple CNN with two convolutional layers, using ReLU as the activation function for the hidden layers and Softmax for the last layer's activation function. The learning rate was set at $\text{lr} = 1 \times 10^{-4}$, and the batch size used was 128. Unless otherwise stated, five clients were uniformly assigned to each group, with each client having 50 data points per class. We've assumed that there are 400 unlabeled public datasets per class. We conducted experiments with 25 local training epochs for the client and 40 distillation epochs to learn from the aggregated logit.

In Section 3.2, we compared the clustering performance of our algorithm with that of an existing FL using clustering. In Section 3.3, we compare the performance of each client after the entire training process with the existing federated distillation algorithm.

4.2. Clustering Experiment

Baseline: We compared our clustering performance to the most commonly used FL method with clustering, Clustered FL (CFL) [14]. CFL uses the cosine similarity of weight updates between clients as a clustering criterion.

Metric: We measured clustering performance using the adjusted rand index (ARI) [22], which quantifies the similarity between true and predicted cluster assignments based on true cluster identity. The silhouette score [23] serves as another metric for evaluating the degree of separation between clusters. The silhouette score was used to assess how well the criterion used for clustering represented the group structure in terms of the logit and similarity of weight updates. For both metrics, higher values indicate better-defined clusters, and the score ranges from -1 to 1.

Hyperparameter: The Table 1 displays the average ARI for various distance thresholds, a hyperparameter in agglomerative clustering. The distance threshold determines how far away from the aggregate clustering values are judged to belong in the same group. We employed two and five for the class per group and four for the number of groups. These results average over each distance threshold value. For future experiments, we choose a distance threshold of two. This is because the range of distance thresholds from 1.5 to 2.5 consistently yields high ARI values above 0.95, indicating optimal clustering.

Table 1. Average adjusted rand index (ARI) for different distance thresholds in agglomerative clustering.

	Distance Thresholds								
	0.25	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0
ARI	0.31	0.67	0.93	0.98	1	1	0.78	0.75	0.43

Silhouette Score: Table 2 illustrates the silhouette scores of the baseline for each clustering method under different group structures, indicating how well the data are clustered. The count vector achieved silhouette scores above 0.5 in all cases, while the model update similarity scores were consistently close to zero, indicating little to no group structure. For both variables, fewer groups led to clearer group structures.

Table 2. Silhouette scores for each clustering criterion, where "Ours" represents the silhouette score for the count vector distribution used by our algorithm and "CFL" represents the silhouette score for the similarity of weight update distribution used by Clustered Federated Learning.

Group	Class per Group							
	2		3		4		5	
	CFL	Ours	CFL	Ours	CFL	Ours	CFL	Ours
2	0.06	0.82	0.03	0.85	0.02	0.81	0.03	0.85
4	0.03	0.88	0.02	0.83	0.01	0.61	0.01	0.78
6	0.02	0.78	0.01	0.77	0.01	0.57	0.01	0.75
8	0.02	0.79	0.01	0.69	0.01	0.60	0.01	0.74
10	0.01	0.76	0.02	0.57	0.00	0.54	0.00	0.72

ARI: Table 3 shows the performance of our algorithm compared to CFL in terms of clustering accuracy. Our algorithm consistently achieved an adjusted rand index (ARI) greater than 0.9 across

different settings, indicating high clustering accuracy. By contrast, CFL recorded an ARI close to 0 in all test cases, demonstrating its persistent ineffectiveness in label-based clustering.

Table 3. Comparison of ARI scores under various group structures for our method and clustered federated learning (CFL).

Group	Class per Group							
	2		3		4		5	
	CFL	Ours	CFL	Ours	CFL	Ours	CFL	Ours
2	-0.03	1.00	0.01	1.00	-0.08	1.00	0.13	1.00
4	0.08	1.00	-0.01	1.00	0.06	0.90	-0.03	1.00
6	0.03	0.96	-0.03	1.00	-0.01	0.96	0.03	1.00
8	-0.03	1.00	0.04	1.00	-0.01	0.93	-0.01	1.00
10	0.01	0.91	0.01	0.93	-0.00	0.97	-0.01	1.00

Existence of Minor Classes: Thus far, our experiments have focused on scenarios where clients in each group have only a subset of the classes. So the boundaries between groups were clear and clustering was relatively easy. However, in this setup, we assumed that three classes per group appeared in large numbers, whereas the remaining classes appeared in smaller numbers. We refer to these infrequent classes as minor classes. We examined the effect of increasing the proportion of minor classes in each group from 5 % to 50 % . The total number of data for each client is 500. For example, if there is a 5% minor class, each client has 25 pieces of data belonging to 7 minor classes and the remaining 475 data belong to three ‘major’ classes. As shown in Table 4 and Figure 2, the silhouette score decreased as the noise class increased and the group structure became less obvious. ARI, on the other hand, was 100% accurate until the noise class reached 30% and then saw a sharp drop in accuracy at 40%.

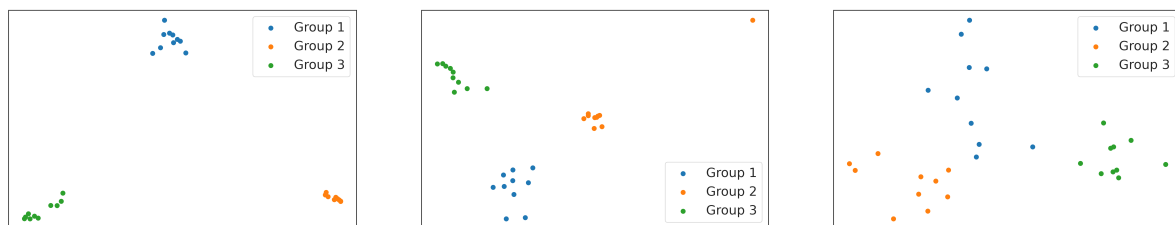


Figure 2. Visualization of the count vector distribution with noise data that has a small percentage of each group. (a) When the noise data rate is 5%. (b) When the noise data rate is 10%. (c) When the noise data rate is 40%.

Table 4. Clustering performance with ‘minor classes’ - the less frequent classes in each group. The title of each row is the proportion of the seven minor classes in each group.

	5%	10%	20%	30%	40%	50%
Silhouette	0.87	0.69	0.59	0.49	0.37	0.33
ARI	1.0	1.0	1.0	1.0	0.9	0.49

4.3. Performance Evaluation

Baseline: In this section, we compare the performance of our method with two baselines: FedDF and DS-FL. FedDF [17] has a similar distillation process to our method, except that it assigns the same logit to all clients. On the other hand, DS-FL (Distillation-Based Semi-Supervised Federated Learning) [24] uses entropy reduction averaging for model output aggregation. It is designed to deliberately reduce the entropy of the global logit before distributing it to mobile devices.

Balanced Group Structure: Table 5 shows that our algorithm consistently outperforms FedDF when the number of clients in each group is equal. When each group contained fewer classes, there were fewer overlapping classes. This resulted in a more distinct group structure. Consequently, our method performed 15% better than the FedDF when there were five classes per group. When the group structure is the clearest, with only two classes per group, our method performed 75% better than the FedDF.

Table 5. Average accuracy comparison between the FedDF method and our proposed group-based distillation method across different group structures.

Group	Class per Group							
	2		3		4		5	
	FedDF	Ours	FedDF	Ours	FedDF	Ours	FedDF	Ours
2	70.0	92.3	74.0	90.9	80.2	90.7	83.2	93.7
4	55.0	98.0	49.8	90.9	70.8	92.9	80.7	92.3
6	30.9	93.8	72.7	93.6	74.6	88.6	69.6	92.8
8	58.9	94.8	58.2	93.3	67.6	87.6	82.8	91.6
10	53.0	90.5	73.5	94.0	81.0	91.7	83.2	92.1
Avg	53.6	93.9	65.6	92.5	74.8	90.3	79.9	92.5

Unbalanced Group Structure: Table 6 and Figure 3 show the performance when the number of clients varies between groups. In the case of global distillation, the accuracy of groups with fewer clients tends to decrease significantly and sometimes approaches zero. Realistically, in this situation, a small number of groups will have to use their own data without applying FL. However, they won't be able to take advantage of the performance gains from FL at all. By contrast, our method ensures that clients in each cluster perform equally well, leading to a significant increase in accuracy for minority groups. This means that all clients can share in the profits of the FL.

Table 6. Performance metrics across groups with different ratios. The 'Group Ratio' column shows the percentage of clients in each group. 'Group Acc' represents the average accuracy achieved by each group at the end of the training process, while 'Total Acc' represents the average accuracy across all clients. The order of the groups in 'Group Acc' and 'Group Ratio' is the same. All values are expressed as percentages.

Group Ratio	Group Acc		Total Acc	
	Ours	FedDF	FedDF	Ours
70, 30	97, 0	96, 92	68	95
80, 20	97, 0	97, 86	77	95
50, 30, 20	96, 1, 0	97, 89, 97	49	94
60, 20, 20	96, 0, 0	96, 91, 93	58	94
40, 30, 20, 10	96, 7, 4, 64	96, 89, 94, 97	48	94
50, 20, 20, 10	96, 0, 0, 66	96, 93, 91, 96	54	94

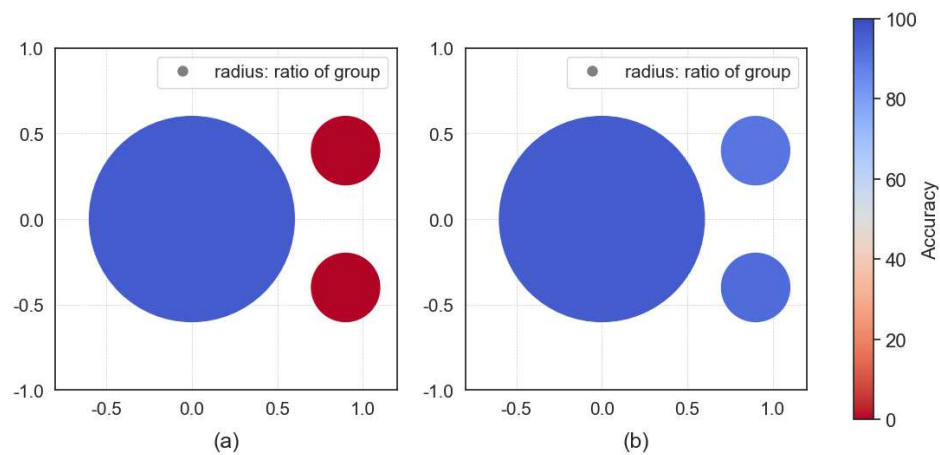


Figure 3. Visualization of accuracy for unbalanced groups, corresponding to the fourth row in Table 6. The radius of each circle indicates the share of the total clients within each group, while the color represents the accuracy of each group. (a) with FedDF and (b) with our method.

Insufficient data: In FL, the amount of data is often insufficient [1]. If a client has insufficient data, it will struggle to train its model effectively. Conversely, a lack of public data interferes with the transfer of the model to the server. We conduct experiments in both client and public data scarce environments. We use 50, 100, 300, and 500 data points for client data, and 100, 300, 500, and 1000 for public data. Figure 4 shows that the performance of other algorithms decreases as the size of the public dataset decreases, while our method maintains its performance. Our algorithm also has the highest accuracy and the lowest variability, represented by the short vertical lines for each data point.

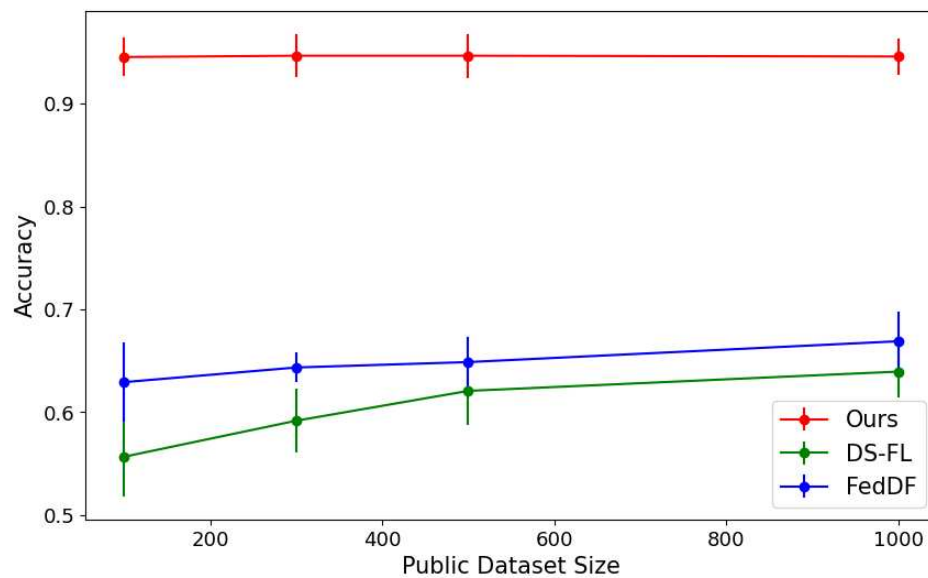


Figure 4. Performance of each algorithm as the amount of public dataset changes.

CIFAR dataset: We use the CIFAR-10 dataset [25], a more complex dataset than MNIST. The CIFAR dataset consists of 60,000 32x32 color images divided into 10 different classes. Our experiments include two and four groups, and two to five classes per group. Figure 5 shows the results, averaged over the number of classes per group. We find that our method becomes more accurate as the number of classes per group decreases.

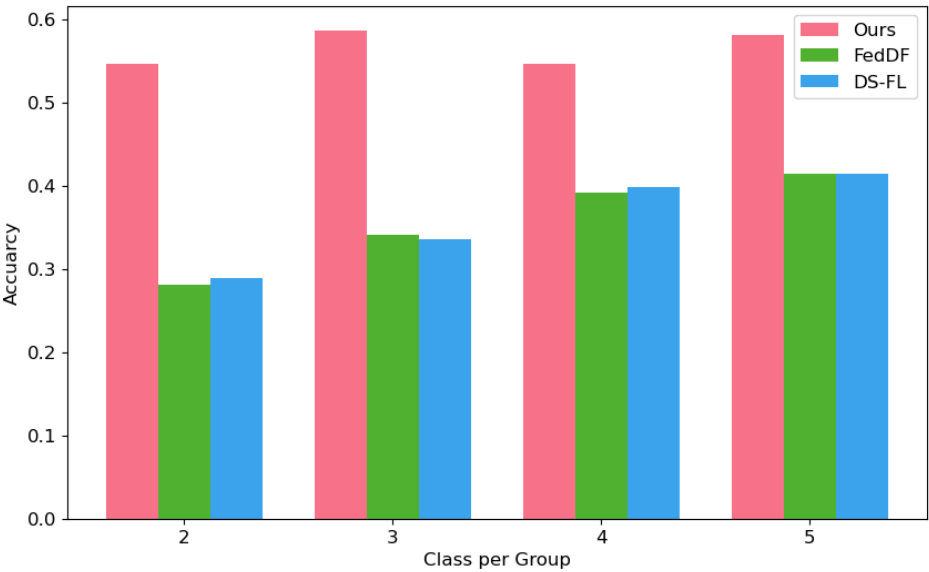


Figure 5. Graph representing accuracy in CIFAR with respect to the class per group.

Existence of Minor Classes: In real-world scenarios, it is often difficult to make clear distinctions between different groups. To address this, we introduce the concept of a "minor class", a less prevalent class within each group, to blur traditional group boundaries. Figure 6 shows that as the proportion of these minor classes increases, the performance gap between our proposed method and traditional approaches narrows.

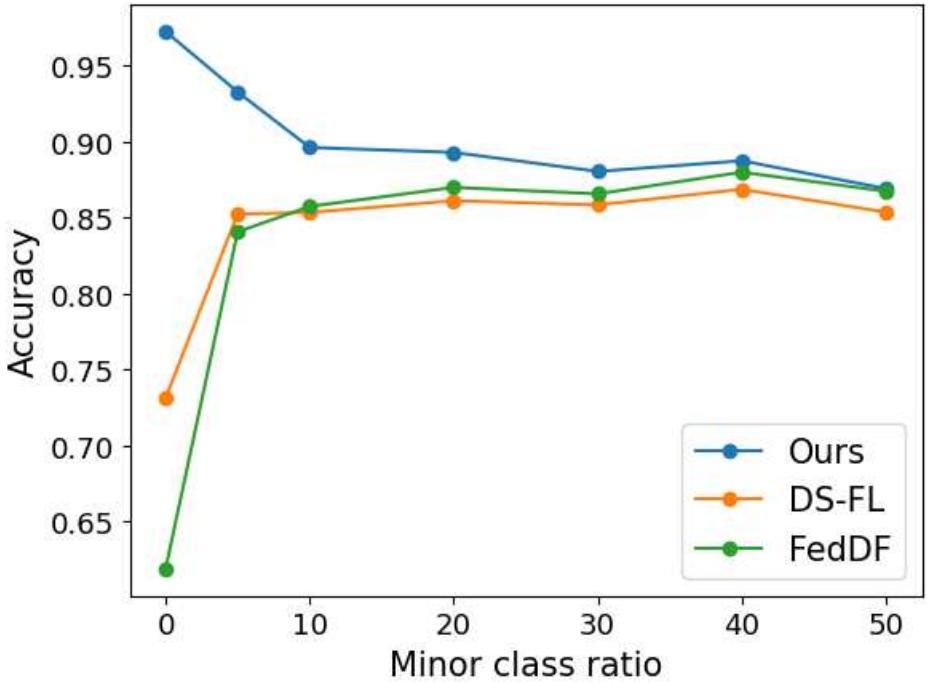


Figure 6. Graph illustrating performance variation as a function of the minor class ratio. Each data point represents the average performance across dataset sizes of 100, 300, 500, and 1000.

5. Discussion

In this study, we address the scenario of different data distributions between different client groups in federated distillation. We introduce a methodology that uses hierarchical clustering to

categorize clients according to the number of labels predicted by each model for public data. This approach overcomes the limitations of traditional federated distillation techniques that assume a uniform data distribution when a label-based group structure exists. Our method can be used when different groups (e.g., demographic groups) have significantly different data distributions to ensure that all groups receive equally good results.

Experiments show that the model correctly classifies groups with different labels. The accuracy of the model exceeds that of traditional methods when there is a clear cluster structure based on labels. In particular, the accuracy of a small number of groups, which is problematic in traditional federated distillation, is significantly improved. This may pave the way for fair FL. Furthermore, our method does not require knowledge of the number of clusters, making it applicable in a wider range of environments. However, as the group structure becomes less clear, the performance gap between our method and existing algorithms narrows. We will continue to improve our method to perform better in ambiguous group structures.

It would be an interesting research topic to combine our method with different data types, such as text, more complex images, or time series data. Our method could also be combined with data-free distillation where no public data exists. Our algorithm will also be very effective in the presence of malicious clients that send false predictions to the server. By creating a group of malicious clients, we can ensure that other clients are not affected by them.

Author Contributions: Conceptualization, Geonhee Yang; Methodology, Geonhee Yang; Software, Geonhee Yang; Validation, Geonhee Yang; Formal Analysis, Geonhee Yang; Investigation, Geonhee Yang; Resources, Geonhee Yang; Data Curation, Geonhee Yang; Writing – Original Draft Preparation, Geonhee Yang; Writing – Review & Editing, Geonhee Yang and Hyunchul Tae; Visualization, Geonhee Yang; Supervision, Hyunchul Tae; Project Administration, Hyunchul Tae; Funding Acquisition, Hyunchul Tae. All authors have read and agreed to the published version of the manuscript.

Funding: This study was carried out with the support of ‘R/&D Program for Forest Science Technology (Project No. 2021383A00-2323-0101)’ provided by Korea Forest Service(Korea Forestry Promotion Institute). This work was supported by the Korea Institute of Industrial Technology as “Development of holonic manufacturing system for future industrial environment [KITECH EO-230006].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be made available on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* **2021**, *14*, 1–210.
2. Zhang, C.; Xie, Y.; Bai, H.; Yu, B.; Li, W.; Gao, Y. A survey on federated learning. *Knowledge-Based Systems* **2021**, *216*, 106775.
3. Rieke, N.; Hancox, J.; Li, W.; Milletari, F.; Roth, H.R.; Albarqouni, S.; Bakas, S.; Galtier, M.N.; Landman, B.A.; Maier-Hein, K.; others. The future of digital health with federated learning. *NPJ digital medicine* **2020**, *3*, 119.
4. Jeong, E.; Oh, S.; Kim, H.; Park, J.; Bennis, M.; Kim, S.L. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479* **2018**.
5. Li, D.; Wang, J. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581* **2019**.
6. Chang, H.; Shejwalkar, V.; Shokri, R.; Houmansadr, A. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. *arXiv preprint arXiv:1912.11279* **2019**.
7. Qayyum, A.; Ahmad, K.; Ahsan, M.A.; Al-Fuqaha, A.; Qadir, J. Collaborative federated learning for healthcare: Multi-modal covid-19 diagnosis at the edge. *IEEE Open Journal of the Computer Society* **2022**, *3*, 172–184.

8. Huang, L.; Shea, A.L.; Qian, H.; Masurkar, A.; Deng, H.; Liu, D. Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. *Journal of biomedical informatics* **2019**, *99*, 103291.
9. Ghosh, A.; Chung, J.; Yin, D.; Ramchandran, K. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems* **2020**, *33*, 19586–19597.
10. Mansour, Y.; Mohri, M.; Ro, J.; Suresh, A.T. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619* **2020**.
11. Briggs, C.; Fan, Z.; Andras, P. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. *IEEE*, 2020, pp. 1–9.
12. Long, G.; Xie, M.; Shen, T.; Zhou, T.; Wang, X.; Jiang, J. Multi-center federated learning: clients clustering for better personalization. *World Wide Web* **2023**, *26*, 481–500.
13. Duan, M.; Liu, D.; Ji, X.; Liu, R.; Liang, L.; Chen, X.; Tan, Y. FedGroup: Efficient clustered federated learning via decomposed data-driven measure. *arXiv preprint arXiv:2010.06870* **2020**.
14. Sattler, F.; Müller, K.R.; Samek, W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems* **2020**, *32*, 3710–3722.
15. Huang, W.; Ye, M.; Shi, Z.; Li, H.; Du, B. Rethinking federated learning with domain shift: A prototype view. *IEEE*, 2023, pp. 16312–16322.
16. He, C.; Annavaram, M.; Avestimehr, S. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in Neural Information Processing Systems* **2020**, *33*, 14068–14080.
17. Lin, T.; Kong, L.; Stich, S.U.; Jaggi, M. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems* **2020**, *33*, 2351–2363.
18. Guha, N.; Talwalkar, A.; Smith, V. One-shot federated learning. *arXiv preprint arXiv:1902.11175* **2019**.
19. Li, Q.; He, B.; Song, D. Practical one-shot federated learning for cross-silo setting. *arXiv preprint arXiv:2010.01017* **2020**.
20. Zhou, Y.; Pu, G.; Ma, X.; Li, X.; Wu, D. Distilled one-shot federated learning. *arXiv preprint arXiv:2009.07999* **2020**.
21. LeCun, Y. The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist/> **1998**.
22. Rand, W.M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association* **1971**, *66*, 846–850.
23. Rousseeuw, P.J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* **1987**, *20*, 53–65.
24. Itahara, S.; Nishio, T.; Koda, Y.; Morikura, M.; Yamamoto, K. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Transactions on Mobile Computing* **2021**, *22*, 191–205.
25. Krizhevsky, A.; Hinton, G.; others. Learning multiple layers of features from tiny images **2009**.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.