Geometry-Aware Aggregation for Federated Learning: Down-Weighting Label-Corrupted Clients via Latent Class Prototypes

Brahim CHEIKH SOUFI[1], Ridha Ejbali[2]
Research Unit Systems and Digital Applications (USDA) of FST, University of Nouakchott, Mauritania, Mauritanie[1]
Research Team in Intelligent Machines (RTIM), National Engineering School of Gabes, University of Gabès, Tunisia[2]

***Abstract*—FedAvg primarily weights client updates based on the volume of data for each participant, without considering the reliability of local labels. This approach can be problematic when a large client has unreliable labels, as its contribution remains significant only because of the number of samples it contains. To address this issue, we introduce Geometric Similarity Weighted Aggregation (GSWA), which combines the number of samples with information derived from the geometric properties of the clients' latent representations. Each client is represented by class-conditional prototypes generated from its latent features. For each class, these prototypes are evaluated against a coordinate-by-coordinate median reference created from the participating clients. The greater the differences between the prototypes, the lower the similarity factors. In this study, we evaluated GSWA on a federated simulation using the scikit-learn digits dataset, distributed across six clients via a Dirichlet partition to create non-independent and non-identically distributed (non-IID) data. When the labels of the most important client were randomly reassigned, GSWA achieved a final accuracy of 96.13% ± 1.35%, compared to 85.82% ± 9.43% for FedAvg over five random initializations. With correct labels, both methods produced similar average accuracies: 97.20% for GSWA and 97.29% for FedAvg. In the evaluated context, GSWA significantly reduced the influence of the corrupted client while maintaining performance close to that of FedAvg when the labels were correct.**

***Keywords*—Federated learning; label noise; non-IID data; class prototypes; robust aggregation; representation learning**

I. Introduction

Federated learning (FL) allows multiple clients to train a shared model while keeping their local training data decentralized [1]. In FedAvg, the server aggregates locally trained models using weights proportional to the number of samples available at each client [1]. This rule reflects client dataset size, but it does not explicitly account for the reliability of the local labels or model updates. This distinction becomes important because client heterogeneity may arise from more than one source. One source is statistical heterogeneity, where clients follow different data distributions, a problem widely studied in federated learning [2]–[4]. The second concerns unreliable contributions — a mislabeled batch, a malfunctioning sensor, a site with different annotation conventions. Dedicated methods address heterogeneous label noise [5], [6], but FedAvg itself has no built-in mechanism for identifying these contributions.

This paper follows a representation-based perspective: instead of aggregating only model parameters, one can characterize each client by the geometry of the latent representations it produces [7], and use that geometry to inform its aggregation weight. Our contribution is twofold. First, we formalize an aggregation rule, GSWA, built on the distance between each client's class prototypes and a cross-client reference obtained through a coordinate-wise median [8]. Second, we evaluate it in a controlled setting and show that it mitigates the impact of a client with corrupted labels in the evaluated scenario — without supplying the aggregation rule with the client's identity or the corruption mechanism.

II. Related Work

Federated learning and heterogeneity. FedAvg was introduced as a communication-efficient method for decentralized learning [1]. Its sensitivity to non-IID data has since been characterized in detail, often via Dirichlet-based partitions that let researchers dial heterogeneity up or down [2], and several surveys catalogue the mitigation strategies that have emerged [3], [4].

Robust federated learning and clustering**.** Some federated learning methods improve robustness by limiting the contribution of clients whose updates differ substantially from the rest. Coordinate-wise median and Krum are representative examples developed for Byzantine-resilient aggregation [8], [9]. Other methods partition clients into groups and learn separate models for clients with similar characteristics [10].

GSWA borrows the median-based reference construction but applies it to representation geometry rather than to raw weights or gradients; the clustering visualization in this study is diagnostic and is not part of the aggregation rule.

Federated learning with label noise. FedCorr identifies noisy clients and corrects labels through a multi-stage procedure [5]. FedNoRo addresses class imbalance and heterogeneous label noise using per-class loss indicators, knowledge distillation, and distance-aware aggregation [6]. GSWA instead investigates weighting local models using distances between class-conditional latent prototypes and a coordinate-wise median reference; the present experiments do not establish superiority over these methods.

Prototypes in federated representation learning. FedProto exchanges class prototypes instead of full model weights,

primarily to cope with heterogeneous architectures and data across clients [7]. We reuse the class-prototype idea, but for a different purpose: not to replace weight sharing, but to audit it — using prototypes as a diagnostic signal that feeds back into the aggregation weights. Related approaches include FedRFQ, which addresses prototype quality and Byzantine-fault-tolerant prototype aggregation [11], and FedSPA, which uses similarity-based prototype aggregation for heterogeneous clients [12]. These approaches provide relevant context for assessing the distinctiveness of the GSWA weighting rule.

Information geometry and learning. Information geometry treats spaces of probability distributions as Riemannian manifolds, equipped with metrics such as the Fisher information metric [13]. The Wasserstein distance, rooted in optimal transport, offers a different metric between distributions and admits a closed form in the Gaussian case [14], [15]. These geometric concepts provide a useful background for comparing client representations. However, the current GSWA implementation follows a simpler formulation: client discrepancies are measured using Euclidean distances between class prototypes. Neither the Fisher–Rao metric nor the full Gaussian Wasserstein distance is used in the aggregation procedure.

III. Theoretical Background

Consider a network split into a feature extractor f and a classification head g, so that $z = f(x)$ is a latent representation and $\hat{y} = g(z)$ the prediction. For a client k holding data (X_k, Y_k), write $Z_k = f_{\theta_k}(X_k)$ for the representations produced by its locally trained model. Rather than comparing clients by their parameters θ_k alone, we consider summaries of the distribution $P_k(Z)$ induced by their local extractors over their own data.

We adopt a simple first-moment summary: rather than modeling the full distribution $P_k(Z)$, each client is reduced to its class prototypes — the mean latent representation conditioned on each observed local label:

$$\mu_{k,c} = \frac{1}{n_{k,c}} \sum_{x \in X_{k,c}} f_{\theta_k}(x), \qquad n_{k,c} > 0. \tag{1}$$

Here $X_{k,c}$ contains the samples assigned observed label c at client k, $n_{k,c} = |X_{k,c}|$, $C_k = \{c : n_{k,c} > 0\}$, and $n_k = \sum_{c \in C_k} n_{k,c}$. Prototypes are computed only for classes present locally. The comparison uses a common architecture and a shared initialization at each round, but subsequent local training may still introduce representation misalignment.

This retains the first moment of $P_k(Z \mid Y = c)$ without requiring a Gaussian distribution. Per-dimension variance does not enter the final weighting scheme. The choice to condition on class labels, rather than compare marginal distributions, was motivated by preliminary comparisons with marginal Gaussian Wasserstein summaries (Eq. (2)). Randomizing labels leaves the empirical distribution of X unchanged. For a fixed extractor f, it therefore also leaves the marginal representation distribution unchanged, whereas class-conditional means depend explicitly on the observed labels. However, when f is trained using those labels, as in GSWA, label corruption can also change the marginal representation distribution. Thus, marginal statistics are not universally blind to label corruption, and the relative benefit of class-conditional comparisons requires a controlled ablation.

$$\begin{aligned} W_2^2(\mathcal{N}_1, \mathcal{N}_2) &= \|\mu_1 - \mu_2\|_2^2 \\ &\quad + \mathrm{Tr}[\Sigma_1 + \Sigma_2 \\ &\qquad -2\big(\Sigma_2^{1/2}\Sigma_1\Sigma_2^{1/2}\big)^{1/2}\Big], \\ \mathcal{N}_i &= \mathcal{N}(\mu_i, \Sigma_i), \quad i \in \{1, 2\}. \end{aligned} \tag{2}$$

Eq. (2) is included for reference [15]. For diagonal covariances with variance vectors v_1 and v_2, it reduces to the sum of squared mean differences and squared standard-deviation differences:

$$W_2^2 = \sum_i \left[(\mu_{1,i} - \mu_{2,i})^2 + (\sqrt{v_{1,i}} - \sqrt{v_{2,i}})^2\right].$$

This marginal Gaussian distance is not used to compute the GSWA aggregation weights, which are defined by Eqs. (3)–(5).

IV. Proposed Method: GSWA

A. A Robust Cross-Client Reference

At every communication round, once local training is done, each client k supplies its class prototypes $\{\mu_{k,c}\}$, their class identifiers, and the counts $n_{k,c}$ required by Eq. (4). For each class c, the server computes a reference prototype as the coordinate-wise median of the prototypes reported by clients that contain that class:

$$\widetilde{\mu}_c = \mathrm{median}_k \left\{\mu_{k,c} : c \in \mathrm{classes}(k)\right\}. \tag{3}$$

The median is used because it is less sensitive to extreme prototype values than an arithmetic mean and is commonly associated with robust aggregation [8]. Its robustness, however, depends on the number of clients representing each class and on whether reliable clients form the majority. A class observed at only one client produces a self-reference, while two contributing clients provide limited protection against a single abnormal prototype. Therefore, the median reference should not be interpreted as guaranteeing either proximity to the true population representation or automatic identification of corrupted clients.

B. Divergence Score and Weighting

For each client k, we measure how far its class prototypes deviate from the corresponding cross-client references. The client-level divergence score is defined as the sample-count-weighted mean of these Euclidean distances over the classes available locally:

$$d_k = \frac{\sum_{c \in C_k} n_{k,c} \left\|\mu_{k,c} - \widetilde{\mu}_c\right\|_2}{n_k}. \tag{4}$$

We transform d_k into a similarity score through an exponential decay. The scale parameter is chosen as the

median divergence across clients in the current round, $\tau = \mathrm{median}_k(d_k)$. This makes the weighting relative to the current client population rather than to a fixed threshold:

$$s_k = \exp\left(-\frac{d_k}{\tau}\right), \qquad w_k = \frac{n_k s_k}{\sum_j n_j s_j}, \qquad \tau > 0. \quad (5)$$

The final weight combines the client sample count with its geometric agreement with the reference. If all similarity factors are equal and positive, GSWA reduces to FedAvg. Otherwise, the normalized weights depend on the relative similarity of all clients. For a fixed τ, increasing d_k decreases s_k, so clients whose prototypes deviate strongly from the cross-client reference can receive substantially smaller aggregation weights.

C. Algorithm

The procedure in Algorithm 1 is applied at each communication round. After local training, participating clients compute class-conditional latent prototypes and report them with the corresponding class counts. The server constructs the coordinate-wise median references, evaluates each client's divergence, and normalizes the resulting similarity-adjusted sample weights before aggregating the local model parameters. Thus, the aggregation rule uses the same local model updates as FedAvg but changes their relative contributions according to prototype agreement.

Algorithm 1 Geometric Similarity-Weighted Aggregation (GSWA)

Input: K clients, initial global model θ_0, T rounds, E local epochs
for $r = 1$ to T **do**
  for each client k in parallel **do**
    $\theta_k \leftarrow \mathrm{LocalTrain}(\theta_{r-1}, \text{local data } k, E)$
    $Z_k \leftarrow f_{\theta_k}(X_k)$ {latent representations}
    Compute $\{\mu_{k,c}\}$ and $\{n_{k,c}\}$
  end for
  For each class c: $\widetilde{\mu}_c \leftarrow$ median of available $\mu_{k,c}$ [Eq. (3)]
  For each client k: compute d_k, s_k, w_k [Eqs. (4)–(5)]
  $\theta_r \leftarrow \sum_k w_k \theta_k$ {weighted aggregation}
end for
Return: θ_T

D. Overhead

Compared with FedAvg, GSWA introduces additional communication and computation because clients transmit class prototypes and the server computes median references and divergence scores. The computational cost is $O(KCd)$, or $O(CdK \log K)$ when median computation relies on sorting. For the 64–64–10 model used in our experiments, transmitting the prototypes represents approximately 13.3% additional uplink values, excluding counts and metadata. Actual runtime and transmitted bytes were not measured.

V. Experimental Setup

A. Data and Non-IID Partitioning

We used the scikit-learn digits dataset, which contains 1,797 grayscale images of size 8×8 across 10 classes. The data were divided into 75% for training and 25% for stratified testing. The training set was then distributed among six simulated clients using a class-wise Dirichlet allocation with $\alpha = 0.3$ to create non-IID data [2]. Input features were standardized using statistics computed from the training split.

B. Model and Hyperparameters

The model is a single-hidden-layer perceptron (64 inputs $\rightarrow$ 64 ReLU units $\rightarrow$ 10 softmax outputs), implemented and trained entirely in NumPy (mini-batch gradient descent, learning rate 0.1, batch size 16). Each client runs 3 local epochs per round, for a total of 25 communication rounds. Aggregate accuracy results are averaged over 5 independent random seeds, which affect the data partition, weight initialization, and mini-batch ordering. Single-run illustrations are identified separately.

C. Scenarios and Baseline

1) Clean scenario: All six clients hold non-IID but correctly labeled data.

2) Corrupted scenario: The client with the largest share of data has all of its labels randomly reassigned — a stand-in for a site whose annotation pipeline has failed. The reported illustrative client holds 358 samples, or 26.6% of the training data. Random reassignment does not necessarily make every assigned label incorrect.

In both scenarios, GSWA is compared against FedAvg using matched initializations, data partitions, data-ordering seeds, and local-training hyperparameters. The aggregation rule is the intended experimental variable; subsequent local model trajectories may differ because the global models differ.

D. Metrics

- Global model accuracy on the held-out test set, tracked every round.
- Aggregation weight assigned to each client over the course of training.
- Final-round inter-client distance matrix and hierarchical clustering, used as exploratory diagnostics rather than as part of the aggregation algorithm.

VI. Results

A. Convergence

Fig. 1 tracks test accuracy over rounds for seed 0. Across five seeds, clean-scenario final accuracies are similar — 97.29% $\pm$ 0.29% for FedAvg versus 97.20% $\pm$ 0.39% for GSWA (Table I). The observed mean difference is small, but these descriptive statistics do not establish equivalence, absence of bias, or a general guarantee of no performance loss.

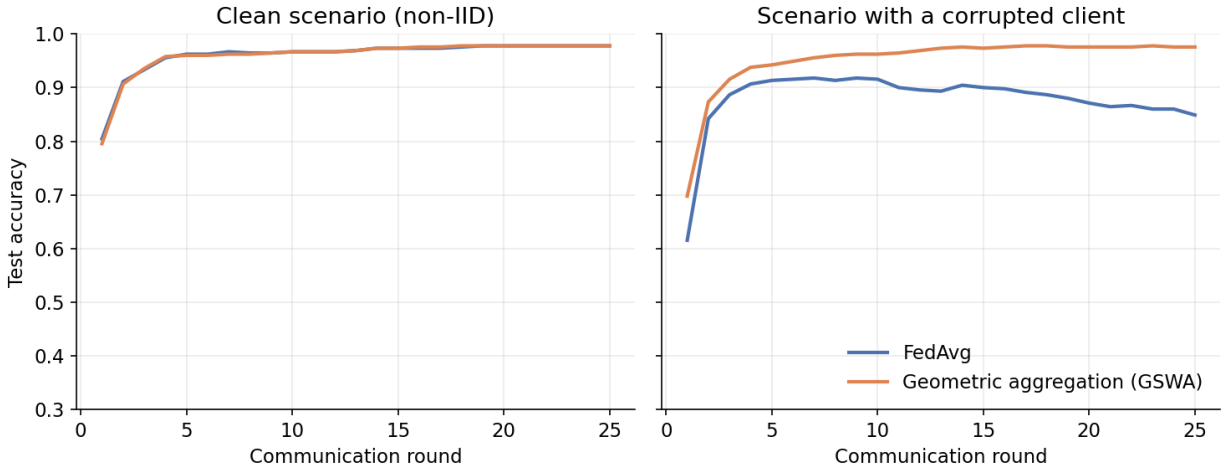

Fig. 1. Test-accuracy convergence curves for the clean and corrupted scenarios, FedAvg vs. GSWA (seed 0).

TABLE I. FINAL ACCURACY (MEAN ± STANDARD DEVIATION OVER 5 SEEDS)

Scenario	FedAvg	GSWA (geometric)
Clean (non-IID, no corruption)	$97.29\% \pm 0.29\%$	$97.20\% \pm 0.39\%$
One corrupted client (random labels)	$85.82\% \pm 9.43\%$	$96.13\% \pm 1.35\%$

In the corrupted setting, FedAvg showed unstable behavior across runs. Its final accuracy averaged 85.82% ± 9.43% over five seeds, whereas GSWA reached 96.13% ± 1.35%. For comparison, GSWA achieved 97.20% ± 0.39% in the clean setting. This improvement should be interpreted only within the specific corruption scenario evaluated here.

B. Down-Weighting the Corrupted Client

Fig. 2 illustrates the evolution of the GSWA weight assigned to the corrupted client. Its weight is already lower than the corresponding FedAvg share of 26.6% at the first reported round and decreases below 1% within five rounds, remaining low afterward. This result shows a strong down-weighting effect in the illustrated run, but it should not be interpreted as a validated binary detection mechanism.

Table II reports the final aggregation weights for seed 0. Under FedAvg, the corrupted client accounts for 26.6% of the total weight, whereas GSWA reduces its contribution to 0.3%. The remaining weight is redistributed according to client sample counts and similarity factors. Minor discrepancies from a total sum of one are due to rounding.

TABLE II. FINAL AGGREGATION WEIGHTS PER CLIENT, CORRUPTED SCENARIO (SEED 0)

Client	C0	C1	C2	C3 (corrupted)	C4	C5
FedAvg weight ($\propto n_i$)	0.077	0.063	0.237	0.266	0.192	0.166
GSWA weight (final)	0.093	0.079	0.337	0.003	0.235	0.254

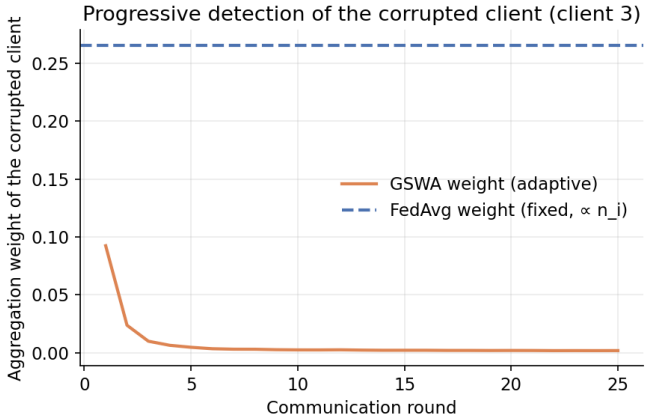

Fig. 2. Aggregation weight assigned to the corrupted client over time: GSWA (adaptive) vs. FedAvg (fixed), illustrative run.

C. Geometric Structure Across Clients

Fig. 3 presents a final-round inter-client distance visualization and hierarchical clustering. In this visualization, the corrupted client C3 has larger displayed distances from the other clients and joins their cluster at a higher linkage height. This is an exploratory illustration; it does not independently validate the GSWA weighting mechanism or establish a detection rate.

D. Summary

Overall, the five-seed results in Table I show similar mean final accuracy for GSWA and FedAvg with clean labels, while GSWA achieves higher mean accuracy and lower variability in the evaluated single-client random-label corruption setting. Fig. 4 summarizes these final accuracies and their standard deviations. The illustrative weight trajectories and client-distance visualization are consistent with down-weighting of the corrupted client in the reported run; they do not establish a general corruption-detection guarantee.

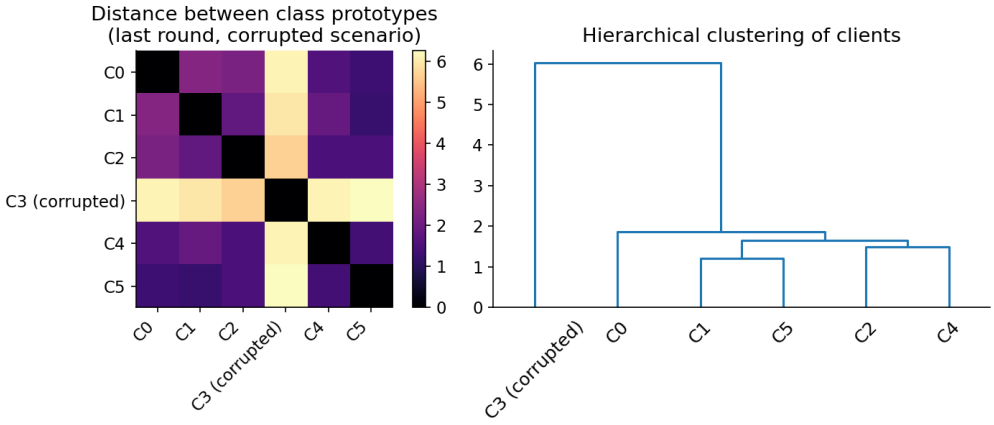

Fig. 3. Inter-client distance visualization (left) and hierarchical clustering (right), corrupted scenario; exploratory diagnostic.

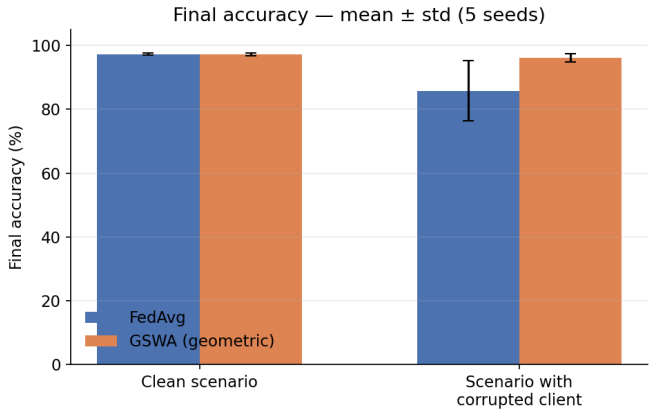

Fig. 4. Final accuracy comparison between FedAvg and GSWA, with error bars (standard deviation over 5 seeds).

VII. Discussion

These results support investigating the geometry of latent representations as information for federated aggregation in addition to sample counts. Class-conditional comparisons depend directly on the relationship between inputs and observed labels, whereas marginal feature statistics do not depend on labels when the extractor is held fixed. Because the extractors here are trained locally, however, label corruption can also affect marginal feature statistics through the learned parameters. A matched ablation is needed before attributing the observed improvement specifically to class conditioning rather than to other differences in the scoring and reference construction.

The proposed mechanism combines a median-based reference with a continuously decreasing exponential similarity factor. It has no explicit detection threshold. Class-specific support, representation comparability, and legitimate distribution shifts can affect the resulting weights. GSWA is related to robust aggregation methods [8], [9], but the present random-label experiment does not establish Byzantine robustness against arbitrary model updates or falsified prototypes.

VIII. Limitations and Threats to Validity

1) Dataset scale: The experiments use the small, low-dimensional digits dataset. The behavior of GSWA on larger datasets, medical images, and deeper architectures such as CNNs or Transformers remains to be evaluated.

2) Corruption setting: We considered only one corruption pattern: random reassignment of all labels at a single client. Partial noise, gradual drift, acquisition bias, multiple corrupted clients, and deliberately falsified updates or prototypes were not examined.

3) Prototype representation: The current weighting rule relies only on class means. Per-class variance and covariance are not used. Extending GSWA with class-conditional Wasserstein information is possible, but its usefulness has not yet been validated, especially when some classes contain few samples.

4) Privacy: GSWA requires sharing class prototypes and sample counts in addition to model updates. Keeping raw data local does not by itself provide a formal privacy guarantee [3], so deployments involving sensitive data would require dedicated privacy and leakage analyses.

5) Baseline coverage: The current study compares GSWA only with FedAvg. Comparisons with coordinate-wise median, Krum, trimmed mean, FedCorr, and FedNoRo are still needed before making broader claims about robustness.

6) Representation alignment and class support: Prototypes obtained from locally updated encoders may not be perfectly aligned across clients. In addition, having a global majority of reliable clients does not guarantee a reliable majority for every class. Rare but valid client distributions may therefore receive lower weights and should be evaluated separately.

IX. Conclusion and Future Work

We proposed and evaluated GSWA, a federated aggregation rule that weights clients according to how well their latent class prototypes agree with a cross-client median reference. On a small prototype, GSWA reduces the influence of a label-corrupted client and improves mean test accuracy relative

to FedAvg in the evaluated scenario, without being given the affected client's identity. Mean performance is similar in the clean scenario. These preliminary results motivate further investigation of representation-based aggregation, but do not establish general corruption detection, Byzantine robustness, or clinical applicability.

The most immediate next steps are: benchmarking against robust and label-noise-aware aggregation methods; validating the prototype-alignment and class-support assumptions; conducting controlled ablations; extending the approach to larger datasets and convolutional architectures; evaluating covariance-aware prototype comparisons; and studying partial or multi-client corruption. Multi-site medical data would constitute a subsequent application study rather than a deployment validated by the present results.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of AISTATS*, ser. PMLR, vol. 54, 2017, pp. 1273–1282.

[2] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

[3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE signal processing magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[4] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-iid data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, 2021.

[5] J. Xu, Z. Chen, T. Q. Quek, and K. F. E. Chong, "Fedcorr: Multi-stage federated learning for label noise correction," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 10 174–10 183.

[6] N. Wu, L. Yu, X. Jiang, K.-T. Cheng, and Z. Yan, "Fednoro: Towards noise-robust federated learning by addressing class imbalance and label noise heterogeneity," *arXiv preprint arXiv:2305.05230*, 2023.

[7] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang, "Fedproto: Federated prototype learning across heterogeneous clients," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 8, 2022, pp. 8432–8440.

[8] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International conference on machine learning*. Pmlr, 2018, pp. 5650–5659.

[9] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[10] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *Advances in neural information processing systems*, vol. 33, pp. 19 586–19 597, 2020.

[11] B. Yan, H. Zhang, M. Xu, D. Yu, and X. Cheng, "Fedrfq: prototype-based federated learning with reduced redundancy, minimal failure, and enhanced quality," *IEEE Transactions on Computers*, vol. 73, no. 4, pp. 1086–1098, 2024.

[12] C. Yang, S. Guo, and P. Zhou, "Fedspa: Heterogenous federated learning with similarity-based prototype aggregation," in *International Conference on Wireless Artificial Intelligent Computing Systems and Applications*. Springer, 2025, pp. 157–169.

[13] S.-i. Amari, *Information geometry and its applications*. Springer, 2016.

[14] N. G. Trillos, M. Jacobs, and J. Kim, "The multimarginal optimal transport formulation of adversarial multiclass classification," *Journal of machine learning research*, vol. 24, no. 45, pp. 1–56, 2023.

[15] D. C. Dowson and B. Landau, "The fréchet distance between multivariate normal distributions," *Journal of multivariate analysis*, vol. 12, no. 3, pp. 450–455, 1982.