Supplementary Material for FedRG: Unleashing the Representation Geometry for Federated Learning with Noisy Clients

Tian Wen[1*] Zhiqin Yang[2*] Yonggang Zhang[4] Xuefeng Jiang[1] Hao Peng[3]
Yuwei Wang[1†] Bo Han[2†]
[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]TMLR Group, Hong Kong Baptist University [3]Beihang University
[4]The Hong Kong University of Science and Technology

Outline

As part of supplementary material for FedRG, we provide further details, organized into the following sections:

- Sec. 1 introduces the implementation details of all experiments.
  - Sec. 1.1 provides the detailed hyperparameter configurations and implementation details for FedRG and all baseline methods
  - Sec. 1.2 gives detailed training settings for the experiments over three datasets.
  - Sec. 1.3 offers a detailed setting for the preliminary experiment shown in Fig. 1 of the main paper.
- Sec. 2 overviews FedRG with its pseudocode.
- Sec. 3 provides additional technical details of the FedRG.
  - Sec. 3.1 provides the definition of the posterior responsibility.
  - Sec. 3.2 presents a diagram illustrating the mapping relationship between semantic clusters and labels.
  - Sec. 3.3 provides the multi-view consistency for semantic augmentations.
  - Sec. 3.4 details the EM update of the vMF geometry model.
  - Sec. 3.5 provides the updating strategy of the class-to-geometry matrix.
- Sec. 4 provides system overhead analysis of FedRG.
  - Sec. 4.1 analyzes the communication overhead.
  - Sec. 4.2 analyzes the computational overhead.
  - Sec. 4.3 analyzes the storage overhead.
- Sec. 5 provides additional experimental results for FedRG.
  - Sec. 5.1 gives additional experimental results on SVHN and CIFAR100 under localized label noise.
  - Sec. 5.2 gives additional experimental results on CIFAR-10 with 100 clients.
  - Sec. 5.3 reports the maximum accuracy across training rounds.
  - Sec. 5.4 compares FedRG with representation- and geometry-related methods.
  - Sec. 5.5 studies the effect of shared self-supervised representations.
  - Sec. 5.6 examines the hyperparameter sensitivity of FedRG.
- Sec. 6 introduces more detailed related works.

Table 1. Hyperparameter settings of FedRG on different datasets.

Hyperparameters	CIFAR-10	CIFAR-100	SVHN
# of training rounds in stage 1 (T_1)	150	150	150
# of training rounds in stage 2 (T_2)	350	350	350
Number of clusters ($\|G\|$)	10	50	10

1. Implementation Details

This section presents the detailed experimental configurations used throughout the paper. We begin by describing the overall hyperparameter settings of FedRG and the baseline methods included in our comparisons. We then elaborate on the main experimental setups reported in the paper, followed by the detailed configurations of the preliminary experiment presented in the main paper.

1.1. Detailed Implementation of Baselines

All baselines listed in this paper can be classified as the general FL methods and the label noise-resilient methods. For the baselines FedAvg [1], FedProx [2], MOON [3], CCVR [4], FedDisco [5], FedSAM [6], FedExp [7], FedInit [8], SymmetricCE [9], FedLSR [10],

*Equal contribution.
†Corresponding author: ywwang@ict.ac.cn,bhanml@comp.hkbu.edu.hk

FedNoRo [11], FedELC [12], and FedCorr [13], we closely follow their official implementations and keep the default hyper-parameter settings whenever applicable.

- FedRG **(Ours)**, consists of two stages. We first perform a 150 rounds of warm-up stage, during which a SimCLR-based network is trained to obtain the label decoupled spherical representation. This is followed by 350 rounds of collaborative training using our representation geometry priority principle. We set the trade-off hyperparameters $\lambda_s = 0.4$ and $\lambda_{\text{n}} = 1$. The number of semantic clusters is fixed for each dataset, as specified in Tab. 1.
- FedProx [2], is proposed to tackle data heterogeneity among clients by adding a fixed proximal term with coefficient μ_{prox} = 0.125 prox to every local loss function.
- MOON [3], mitigates client drift via a contrastive regularizer. We set the temperature $\tau = 0.5$ and the contrastive weight $\mu = 1.0$.
- CCVR [4], calibrates the classifier by estimating classwise feature means and covariances. We set sample_per_class = 100, which determines how many synthetic feature samples are drawn for each class when constructing the virtual representation set.
- FedDisco [5], reweights client updates according to their distributional divergence. We adopt $a = 0.5$ and $b = 0.1$, where a controls the penalization strength on clients with larger divergence, and b provides a small positive offset to stabilize aggregation.
- FedSAM [6], applies Sharpness-Aware Minimization to each client's local update. We use the SAM optimizer with perturbation radius $\rho = 0.1$ and set $\eta = 0$ following the common configuration. Stochastic Weight Averaging (SWA) is enabled with swa_c = 1 and swa_lr = 0.0001.
- FedInit [8], performs relaxed initialization by partially pulling each client's model toward the global model. Specifically, we set $\beta = 0.1$, which controls the interpolation strength in the update $w \leftarrow w_g + \beta(w_g - w_{\text{local}})$ applied at the start of each round.
- SymmetricCE [9], replaces the standard cross-entropy with a symmetric combination of CE and reverse CE. We set $\alpha = 0.1$ and $\beta = 1.0$, where α weights the conventional CE term and β controls the contribution of the reverse CE term.
- FedLSR [10], introduces label-wise soft regularization. Following the original design, we set $\lambda_e = 0.6$ for the entropy-based regularization term and $\gamma = 0.4$ for controlling the strength of the L1 prediction consistency.
- FedNoRo [11], performs two-stage training to mitigate the effect of noisy clients. We set the warm-up round to 10. In the second stage, noisy clients adopt a logit-adjusted knowledge-distillation loss.
- FedELC [12], extends federated learning under noisy labels. In the first stage, FedELC performs 20 rounds of warm-up. For noisy clients, the soft label matrix is initialized with K_{pencil} = 10 on the true label dimension.
- FedCorr [13], identifies and corrects noisy clients through a three-stage procedure. In the first stage, clients train for 5 rounds with MixUp ($mixup_\alpha$= 1.0), and noisy clients are detected via accumulated LID and a loss-based GMM. Detected noisy clients are relabeled using confidence filtering with confidence threshold of 0.5, and relabel ratio of 0.5, while a proximal term with β = 5 stabilizes updates. In the second stage, clean clients fine-tune the model for 250 rounds. Finally, the third stage resumes standard FL for 245 rounds.
- FedClean [14], adopts a three-stage framework to progressively identify and mitigate noisy labels. During the warm-up phase (100 rounds), clients are trained with standard cross-entropy to obtain reliable confidence estimates. In stage 1 (200 rounds), each client separates clean and noisy samples using confidence-based filtering. Stage 2 (200 rounds) further refines the model by training on the cleaned dataset.
- FedProto [15], transfers client knowledge through class prototypes in the feature space. We use one local epoch, standard cross-entropy training, EMA decay 0.9, and weighted client aggregation, following its original formulation.
- FedCNI [16], identifies noisy samples through supervised centroid consistency. We use one local epoch, centroid-consistency weight λ sim = 0.7, GMM threshold 0.5.
- GGEUR [17], improves robustness with geometry-aware feature augmentation based on local class statistics. We use augmentation weight aug = 0.5, three augmented features per real feature, top-20 eigen-directions, covariance ridge 1e-4, and minimum class count 20.
- FedROM [18], adopts a representation-oriented robust objective with a warm-up stage and an additional OT regularizer. We use 50 warm-up rounds, and the original SCE setting with $\alpha = 0.1$ and $\beta = 1.0$.

Table 2. List of datasets used in our experiments.

Hyperparameters	**CIFAR-10**	**CIFAR-100**	**SVHN**
Size	50,000	50,000	73,257
# of classes	10	100	10
Batch size	64	64	64
# of clients	10 / 100	10	10
Selected fraction	0.5 / 0.1	0.5	0.5
Learning rate	0.01	0.01	0.01
Weight decay	0.0005	0.0005	0.0005
Local epochs	1 / 5	1	1
Heterogeneity	0.1	0.1	0.1
Architecture	ResNet-18	ResNet-34	ResNet-18

1.2. Experimental Settings over Three Datasets

All experiments are conducted on a server with 8 NVIDIA A100 GPUs. Unless otherwise stated, each client is trained with learning rate 0.01 and momentum 0.9. We also set the batch size to 64, following previous works [12, 13]. Across all experiments, we follow a unified training setup for the three datasets, as summarized in Tab. 2. We adopt SGD with momentum 0.9 and weight decay 0.0005 as the local optimizer, using a batch size of 64 for CIFAR-10/100 and SVHN. Following the prior works [13, 14], the learning rate is set to 0.01 for CIFAR-10/100 and SVHN. All remaining dataset-specific details are provided in Tab. 2.

1.3. Implementation Details for Fig.1 in Main Paper

To support the observations presented in Fig. 1 of the main paper, we conduct a preliminary study that examines the noise identification behavior of different methods under both symmetric and globalized label noise on the CIFAR-10 dataset. The experiment runs for 500 communication rounds with a total of 10 clients, where 50% of the clients are randomly selected at each round. All other training configurations follow the unified setup summarized in Tab. 2.

The compared baselines rely on small loss values to identify noisy samples, which becomes unreliable under severe data heterogeneity. In contrast, `FedRG` utilizes the geometry of the learned spherical representation to distinguish clean and noisy instances more robustly, especially in challenging heterogeneous environments.

2. Pseudocode Overview of `FedRG`

To clearly present the overall workflow of `FedRG`, we provide a concise algorithmic summary in Algorithm 1. The method consists of two major stages. In the first stage, clients collaboratively learn label-decoupled spherical representations through SimCLR-based pre-training, which serves as a noise-agnostic feature extractor. In the second stage, `FedRG` performs geometry-based noisy sample identification using the vMF model, updates class-to-geometry matrix, and optimizes both the local model and the client-specific noise absorption matrix in a robust manner. These steps together enable `FedRG` to handle heterogeneous and instance-independent label noise effectively in federated scenarios.

Algorithm 1 FedRG

Require: Number of clients K, communication rounds T_1, T_2, noisy local datasets $\{\tilde{\mathcal{D}}_k\}_{k=1}^{K}$, initial global model θ

Ensure: Global model θ^{final}

// Stage I: Label-decoupled spherical representation (SimCLR pre-training)

for $t = 1$ to T_1 **do**
  Server samples clients $\mathcal{S}_t \subseteq \{1, \dots, K\}$.
  for each client $k \in \mathcal{S}_t$ in parallel **do**
    Initialize local model $\theta_t^{(k)} \leftarrow \theta_{t-1}$.
    for mini-batch $(x_i, \tilde{y}_i) \subset \tilde{\mathcal{D}}_k$ **do**
      Update $\theta_t^{(k)}$ by minimizing loss $\mathcal{L}_{\text{SimCLR}}$.
    end for
  end for
  Server aggregates $\theta_t \leftarrow \sum_{k \in \mathcal{S}_t} \omega_k \theta_t^{(k)}$ ▷ FedAvg
end for
Initialize vMF model and class-to-geometry matrix.
Initialize noise absorption matrix $\mathbf{T}_k$.

// Stage II: Geometry-based noisy data identification and robust optimization

for $t = T_1 + 1$ to $T_1 + T_2$ **do**
  Server samples $\mathcal{S}_t$ and broadcasts θ_{t-1}.
  for each client $k \in \mathcal{S}_t$ **in parallel do**
    Initialize clean set $\mathcal{D}_k^c \leftarrow \emptyset$, noisy set $\mathcal{D}_k^n \leftarrow \emptyset$.
    for mini-batch $(x_i, \tilde{y}_i) \subset \tilde{\mathcal{D}}_k$ **do**
      Compute geometry–label consistency score $\tilde{q}_i(\tilde{y}_i)$ and clean posterior P_i^{clean}.
    end for
    Fit GMM on $\{1 - P_i^{\text{clean}}\}$ and split data into clean subset $\mathcal{D}_k^c$ and noisy subset $\mathcal{D}_k^n$.
    Update matrix B using only $\mathcal{D}_k^c$ and update the vMF model using all local samples.
    Optimize θ_{t-1} and local $\mathbf{T}_k$ by minimizing $\mathcal{L} = \lambda_s \mathcal{L}_{\text{SCE}}(\tilde{\mathcal{D}}_k) + \lambda_n \mathcal{L}_n(\mathcal{D}_k^n; T_k)$.
    Obtain updated $\theta_t^{(k)}$ and upload to server.
  end for
  Server aggregates $\theta_t \leftarrow \sum_{k \in \mathcal{S}_t} \omega_k \theta_t^{(k)}$.
end for
return θ^{final}

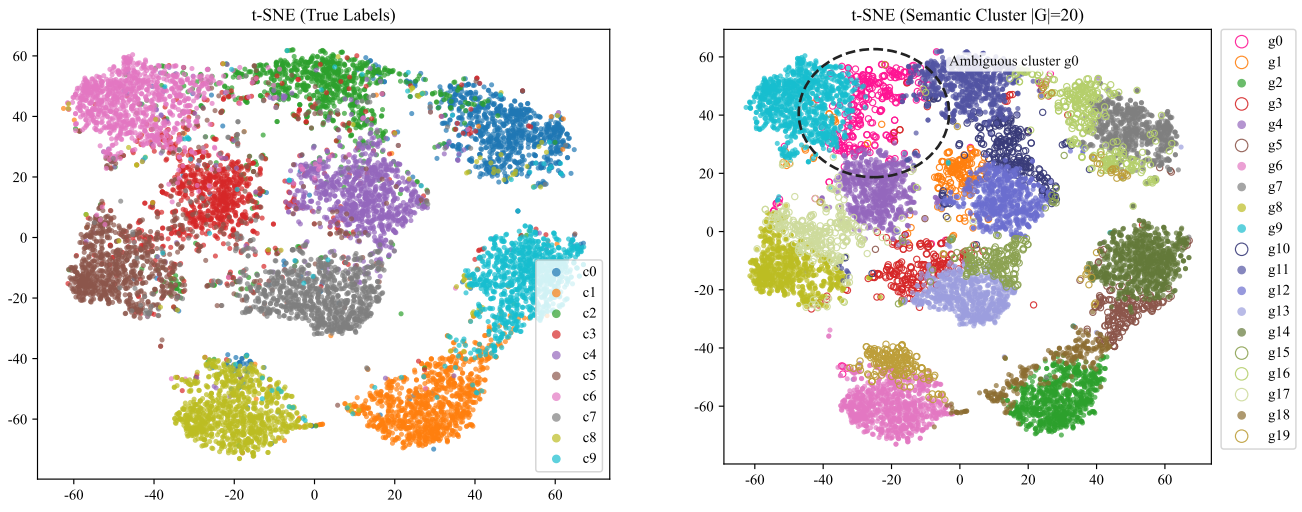
3. Additional Technical Details underlying the `FedRG` Solution

3.1. Definition of Posterior Responsibility

In Sec. 3.4 of the main paper, we model the label decoupled spherical representations with a mixture of semantic clusters on the hypersphere. Given this mixture, the *posterior responsibility* of a cluster for a sample is defined as the posterior probability that this sample belongs to that cluster under the current mixture model. Concretely, let z_i denote the unit-norm representation of sample i, and let $p_g(z_i)$ be the density of cluster g with mixture weight π_g. The posterior responsibility of cluster g for sample i is

$$\gamma_{i,g} = \frac{\pi_g \, p_g(z_i)}{\sum_{h=0}^{G} \pi_h \, p_h(z_i)}, \tag{1}$$

which is a normalized score over components and can be interpreted as a soft assignment of z_i to semantic cluster



(a) Sample classes

(b) Semantic clusters

Figure 1. t-SNE visualization of spherical representations on CIFAR10. (a) colored by ground-truth labels. (b) colored by FedRG semantic clusters. Hollow markers indicate the semantic clusters with multi-class membership, and the dashed bounding box highlights one representative cluster $g0$, which is ambiguous. This cluster spans two neighboring class regions (class $c2$ and $c6$), illustrating that FedRG discovers fine-grained semantic modes and supports soft cluster-to-label associations rather than rigid one-to-one mappings.

g (or to the background when $g = 0$). Rather than committing each sample to a single cluster, the vector $\Gamma_i = (\gamma_{i,1}, \ldots, \gamma_{i,G})$ summarizes how strongly each semantic component is *responsible* for generating the observed representation of that sample.

This probabilistic notion of responsibility plays two key roles in FedRG. First, it provides a *label-free geometric descriptor* of each sample: samples that share similar semantics tend to have similar responsibility profiles over the mixture clusters, even when their annotated labels are noisy or inconsistent across clients. Second, by comparing the label-free responsibilities with the label-conditioned geometry (built from annotated labels), FedRG can quantify how well the observed label agrees with the intrinsic representation geometry. Samples whose posterior responsibilities are concentrated on components that are inconsistent with their annotated label are flagged as likely noisy, while samples whose responsibilities align with the label-conditioned component distribution are treated as clean. In this way, responsibilities serve as the fundamental bridge between the spherical representation manifold and the subsequent clean/noisy identification procedure in FedRG.

3.2. Cluster-to-Geometry Mapping

Geometric relationship between Semantic Clusters and Class Labels. A key property of FedRG is that the relationship between semantic clusters and ground-truth classes is fundamentally **many-to-many**: on the one hand, a single class typically contains multiple fine-grained semantic clusters; on the other hand, a semantic cluster may softly associate with multiple classes through the learned distribution $\beta_{c,g}$. This many-to-many structure underlies the proposed class-to-geometry mapping in FedRG and enables a richer characterization of representation geometry than rigid class-partitioning approaches.

We visualize the learned spherical representations using t-SNE under two coloring schemes. Fig. 1a shows the embedding colored by ground-truth labels, where each class forms a coherent island-shaped region. When the same embedding is colored by the semantic clusters produced by FedRG with $|G| = 20$ clusters (Fig. 1b), a markedly different structure emerges: each class is decomposed into several compact and well-separated sub-clusters. This confirms that categories in federated data are not monolithic; rather, they are composed of multiple latent semantic modes capturing intra-class variation in pose, background, texture, and illumination.

Furthermore, the cluster geometry clearly reveals the soft cross-class association of these semantic components. In Fig. 1b, we mark *ambiguous clusters* with hollow markers. A representative ambiguous cluster, denoted as $g0$ and highlighted by a dashed bounding box, lies precisely across the boundary between two adjacent classes ($c2$ and $c6$) in Fig. 1a. Such cross-class clusters naturally arise in semantically confusing regions and often correspond to borderline samples or pockets of noisy labels. Their existence empirically illustrates that clusters in FedRG can exhibit multi-modal associations over labels, exactly what the $\beta_{c,g}$ is de-

signed to capture.

Taken together, these observations are consistent with the core design of FedRG: instead of enforcing a rigid one-to-one correspondence between clusters and classes, FedRG models the representation geometry with multiple clusters per class and soft class associations per cluster. This structure provides a principled basis for modeling semantic granularity, detecting unreliable samples, and achieving robustness under federated noisy-label settings.

3.3. Multi-view Consistency for Semantic Augmentations

This subsection supplements Sec. 3.4.2 (main paper) by detailing how multi-view self-supervision is used to enhance the stability of posterior responsibilities on the hypersphere. For each training sample x_i, two stochastic augmentations are generated as $z_{i,1},\, z_{i,2} \in \mathbb{S}^{d-1}$, producing unit-normalized representations. Their consistency is summarized by the length of the averaged vector as

$$R_i \;=\; \left\| \frac{z_{i,1} + z_{i,2}}{2} \right\|_2 \in [0,1]. \tag{2}$$

Under spherical statistics, R_i is a sufficient statistic for the concentration of a vMF distribution in dimension d, so we map it to an augmentation induced concentration

$$\hat{\kappa}_i^{\mathrm{aug}} \;=\; A_d^{-1}(R_i), \tag{3}$$

where $A_d(\kappa)$ is the vMF mean resultant length and A_d^{-1} is the inverse. We then compute a relative observation precision by normalizing with respect to the batch-wise median

$$\overline{\kappa}^{\mathrm{aug}} \;=\; \mathrm{median}_j\big(\hat{\kappa}_j^{\mathrm{aug}}\big), \qquad r_i \;=\; \frac{\hat{\kappa}_i^{\mathrm{aug}}}{\overline{\kappa}^{\mathrm{aug}}}. \tag{4}$$

In implementation, the median normalizer is maintained by an exponential moving average across mini-batches, and r_i is clipped to a bounded interval for numerical stability. Given the vMF model, the initial responsibilities for clusters g are

$$\gamma_{i,g} \;\propto\; \pi_g \exp\big(\kappa_g\, \mu_g^\top z_i\big), \tag{5}$$

where π_g is the mixture weight, μ_g the mean direction, and κ_g the concentration of component g. The augmentation-aware observation precision r_i is used to rescale the effective concentration for sample i:

$$\kappa_g^{(i)} \;=\; r_i\, \kappa_g, \qquad \tilde{\gamma}_{i,g} \;\propto\; \pi_g \exp\big(\kappa_g r_i\, \mu_g^\top z_i\big), \tag{6}$$

followed by normalization over all clusters. In implementation, these quantities are instantiated as vMF-style unnormalized energy scores over the foreground components together with a background score, followed by a softmax normalization with a fixed temperature. Samples whose two views are highly consistent (large R_i and thus larger r_i) yield sharper responsibilities, while unstable samples induce softer responsibilities.

3.4. EM Update of the vMF Model

This subsection supplements Sec. 3.5 (main paper) by explaining how the vMF model is updated in an EM-style manner using the responsibilities. Let i index samples in a mini-batch, $g \in \{0, 1, \ldots, G\}$ index vMF clusters (with $g = 0$ the background), and $c \in \{1, \ldots, C\}$ index classes. For each sample x_i, we denote by $\mathbf{z}_i \in \mathbb{R}^d$ its ℓ_2-normalized feature on the sphere, by $\tilde{\gamma}_{i,g}$ the refined responsibility of cluster g, and by $\tilde{\gamma}_i^{\mathrm{bg}} = \tilde{\gamma}_{i,0}$ the background responsibility. The clean probability P^{clean} is converted into a binary clean indicator $m_i \in \{0, 1\}$ via a two-component Gaussian mixture on $1 - P^{\mathrm{clean}}$, where the component with smaller mean is treated as clean. The model parameters are the mean directions $\boldsymbol{\mu}_g \in \mathbb{R}^d$, concentrations $\kappa_g \in \mathbb{R}_+$, cluster weights $\pi_g \in (0, 1)$ with $\sum_{g=1}^{G} \pi_g = 1$, background mass $\pi_0 \in (0, 1)$.

In the E-step, the vMF model provides augmented posteriors over all clusters and the background that satisfy $\sum_{g=1}^{G} \tilde{\gamma}_{i,g} + \tilde{\gamma}_i^{\mathrm{bg}} = 1$. These responsibilities are treated as $p(g \mid x_i)$ in the subsequent updates.

For the M-step on vMF model, we maintain a buffer of sufficient statistics accumulated over mini-batches. For each non-background cluster $g \in \{1, \ldots, G\}$, we store a responsibility-weighted feature sum $\mathbf{s}_g = \sum_i \tilde{\gamma}_{i,g}\, \mathbf{z}_i$, a soft count $n_g = \sum_i \tilde{\gamma}_{i,g}$, and for the background $n_{\mathrm{bg}} = \sum_i \tilde{\gamma}_i^{\mathrm{bg}}$. At a fixed update interval, the buffered statistics are converted into provisional parameters. The updated mean direction of cluster g is

$$\hat{\boldsymbol{\mu}}_g = \frac{\mathbf{s}_g}{\|\mathbf{s}_g\|_2}\,, \tag{7}$$

and the empirical mean resultant length is

$$R_g = \frac{\|\mathbf{s}_g\|_2}{n_g + \varepsilon}\,, \tag{8}$$

where ε is a small constant for numerical stability. The new concentration is obtained via the inverse vMF mean resultant function in dimension d,

$$\hat{\kappa}_g = A_d^{-1}(R_g)\,, \tag{9}$$

and is clipped to a reasonable range in the implementation. The actual parameters are updated by an online EM step with momentum $\rho \in (0, 1)$:

$$\boldsymbol{\mu}_g \leftarrow (1 - \rho)\, \boldsymbol{\mu}_g + \rho\, \hat{\boldsymbol{\mu}}_g\,, \tag{10}$$

$$\kappa_g \leftarrow (1 - \rho)\, \kappa_g + \rho\, \hat{\kappa}_g\,, \tag{11}$$

followed by renormalizing $\boldsymbol{\mu}_g$ to unit length.

The cluster weights and background mass are updated from the soft counts using a Dirichlet-style prior with hyperparameter α_{dp}:

Table 3. Experimental results on SVHN and CIFAR100 with localized noise setting. The **bold** denotes the best result while the underlined denotes the second-best result.

Dataset	**SVHN**						**CIFAR100**					
Noise Type	Symmetric Label Noise			Pairflip Label Noise			Symmetric Label Noise			Pairflip Label Noise		
Metric	Accuracy	Precision	F-score	Accuracy	Precision	F-score	Accuracy	Precision	F-score	Accuracy	Precision	F-score
FedAvg [1]	31.00	34.80	25.25	33.01	36.37	24.59	37.84	38.94	37.26	44.19	46.42	43.75
FedProx [2]	30.77	45.05	22.11	42.89	50.53	33.81	33.83	35.20	33.21	44.10	46.29	43.56
MOON [3]	44.20	**60.55**	35.93	42.22	47.75	35.83	36.40	37.21	35.74	43.36	45.60	42.98
CCVR [4]	30.55	34.64	23.15	34.72	35.89	26.30	37.29	37.74	36.79	44.52	46.70	44.34
FedDisco [5]	29.70	33.69	22.86	32.62	33.77	23.66	37.05	37.99	36.48	43.82	46.06	43.42
FedSAM [6]	35.47	45.72	22.13	40.58	40.14	28.00	42.83	46.51	42.04	50.73	54.14	50.17
FedExp [7]	29.94	32.90	23.44	31.77	30.61	22.31	37.75	38.33	37.20	43.27	45.26	42.90
FedInit [8]	29.54	34.06	24.22	29.53	30.80	21.36	37.41	38.10	36.81	43.93	46.01	43.57
SymmetricCE [9]	49.49	52.94	40.07	41.64	41.15	32.96	53.02	56.52	52.37	43.10	45.85	42.28
FedLSR [10]	37.86	27.95	23.61	40.89	29.83	27.60	14.25	15.42	16.62	13.04	14.52	15.63
FedNoRo [11]	32.25	30.90	29.69	31.72	33.26	27.95	36.26	36.76	35.74	39.84	41.21	39.83
FedELC [12]	43.27	46.24	41.04	30.36	32.45	24.91	38.18	38.12	37.59	31.57	26.57	24.79
FedCorr [13]	45.28	47.28	43.25	39.57	42.25	38.55	38.78	40.20	39.58	35.87	53.19	34.94
FedClean [14]	42.93	40.33	45.48	44.38	40.22	37.87	43.24	40.28	45.22	40.87	42.58	41.27
FedRG	**57.72**	50.92	**47.01**	**55.76**	**56.34**	**48.36**	**53.58**	**56.63**	**52.84**	**57.60**	**62.18**	**56.51**

$$\hat{\pi}_g = \frac{n_g + \alpha_{\mathrm{dp}}/G}{\sum_{h=1}^{G}(n_h + \alpha_{\mathrm{dp}}/G)}, \tag{12}$$

and then smoothed by

$$\pi_g \leftarrow (1-\rho)\,\pi_g + \rho\,\hat{\pi}_g\,, \tag{13}$$

3.5. Updating Strategy of Class-to-Geometry Matrix

For the M-step on the class-to-geometry mapping, only samples currently regarded as clean contribute. Let $\tilde{y}_i$ be the noisy label of x_i. We define the clean-weighted counts

$$N_{c,g} = \sum_i m_i\,\mathbf{1}[\tilde{y}_i = c]\,\tilde{\gamma}_{i,g}\,, \tag{14}$$

where $\mathbf{1}[\cdot]$ is the indicator function. With Dirichlet smoothing parameter α_M, the updated class-to-geometry mappings are

$$\hat{\beta}_{c,g} = \frac{N_{c,g} + \alpha_M}{\sum_{g'=1}^{G}(N_{c,g'} + \alpha_M)}, \tag{15}$$

and are again updated with momentum

$$\beta_{c,g} \leftarrow (1-\rho)\,\beta_{c,g} + \rho\,\hat{\beta}_{c,g}\,. \tag{16}$$

4. System Overhead Analysis

We summarize the overhead of `FedRG` w.r.t. FedAvg from three axes: communication, computation, and storage. To avoid repetition, we only keep the quantitative definitions and the key dominant terms, algorithmic details of Stage I/II follow Sec. 3 in the main paper and Algorithm 1.

Notation. M: number of scalars in model θ (parameter footprint). C: #classes. G: #semantic clusters. d: feature dimension on the hypersphere (CIFAR-10: d=512, G=10). F_f, F_b: per-mini-batch forward/backward cost of the backbone. We denote by Δ_c the extra embedding-level geometry cost, and by Δ_m the extra persistent memory footprint.

4.1. Communication

Under the default `FedRG` used in the main experiments, each selected client only uploads and downloads the model parameters θ (size M), so the per-round communication pattern is identical to FedAvg. If one additionally enables cross-client aggregation of the personalized noise absorption matrix $\mathbf{T}_k$ for the ablation, the extra upload is C^2 scalars per client, corresponding to the additional overhead $\mathrm{Overhead}_{\mathrm{comm}} = \frac{C^2}{M} \times 100\%$, where CIFAR-10/ResNet-18: C^2=100, $M \approx$11.18M $\Rightarrow$ 100/11.18M $\approx 0.0009\%$.

4.2. Computation

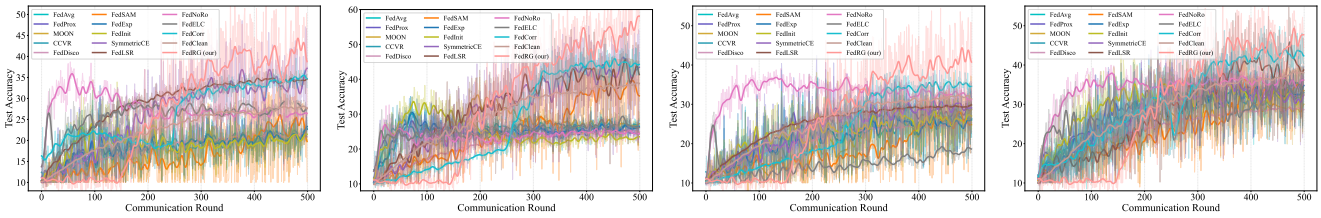
Stage I (SimCLR warm-up) uses two stochastic views per sample, i.e., two forwards and one backward per mini-batch:

$$\#\mathrm{Comp}(\text{Stage I}) = 2F_f + F_b. \tag{17}$$

Stage II keeps the same backbone update (two-view forward with one backward) and adds embedding-level geometry inference:

$$\#\mathrm{Comp}(\text{Stage II}) = 2F_f + F_b + \Delta_c. \tag{18}$$

Here Δ_c is dominated by the geometry-specific routines after feature extraction: (i) a small batched similarity $(B\times d)\cdot(d\times G)$ for tempered vMF responsibilities (CIFAR-10: d=512, G=10), (ii) a 2-GMM on 1D cleanliness scores



(a) Localized symmetric noise (b) Globalized symmetric noise (c) Localized pairflip noise (d) Globalized pairflip noise

Figure 2. Test accuracy versus communication rounds with $K = 100$ clients on CIFAR10 under four label noise configurations. We conduct Stage 1 in the first 150 rounds.

Table 4. Experimental results on CIFAR10, SVHN, and CIFAR100 datasets under four different label noise settings, including LN_1: localized symmetric noise, LN_2: globalized symmetric noise, LN_3: localized pairflip noise, LN_4: globalized pairflip noise. CIFAR10* denotes the CIFAR10 dataset trained with 100 clients. The **bold** denotes the best result while the underlined denotes the second-best result.

Dataset	**CIFAR10**				**SVHN**				**CIFAR100**				**CIFAR10***			
Noise Setting	LN_1	LN_2	LN_3	LN_4	LN_1	LN_2	LN_3	LN_4	LN_1	LN_2	LN_3	LN_4	LN_1	LN_2	LN_3	LN_4
FedAvg [1]	55.68	66.88	57.48	56.88	53.99	77.57	48.56	65.85	42.81	47.27	49.51	43.96	30.24	34.99	38.70	44.69
FedProx [2]	61.10	64.45	62.30	57.73	57.24	71.60	57.78	64.62	40.90	45.06	49.51	43.88	30.88	33.54	42.44	44.29
MOON [3]	68.85	68.80	64.31	60.26	62.60	75.02	65.46	70.99	44.23	47.73	50.45	43.49	31.41	40.55	38.24	36.09
CCVR [4]	68.95	68.90	60.86	56.32	59.74	76.86	51.64	68.67	44.5	47.85	51.42	45.68	28.42	32.57	35.47	42.60
FedDisco [5]	68.85	68.95	58.71	53.36	55.18	77.22	47.68	70.59	42.28	47.43	49.19	43.96	24.16	26.95	26.27	30.47
FedSAM [6]	56.52	67.20	66.48	63.61	45.66	47.78	55.29	85.31	46.86	49.99	53.42	47.43	36.45	46.06	33.01	45.20
FedExp [7]	66.19	66.21	60.25	54.45	55.19	77.39	51.39	67.38	41.78	46.59	49.46	43.60	33.07	35.03	36.58	44.92
FedInit [8]	57.89	67.06	61.55	53.78	53.58	80.85	47.81	67.00	42.68	47.40	50.00	43.41	34.42	38.26	38.06	44.10
SymmetricCE [9]	69.05	71.57	63.71	66.75	82.78	83.68	77.17	81.60	**55.61**	**59.10**	49.91	48.34	48.52	56.68	40.65	45.99
FedLSR [10]	66.21	68.85	40.69	43.55	61.11	82.00	55.06	49.07	15.06	17.01	13.88	19.20	37.39	54.95	30.87	54.73
FedNoRo [11]	64.87	67.82	53.94	50.38	59.96	72.86	53.67	50.32	45.58	45.72	46.08	41.62	38.70	32.67	39.93	39.97
FedELC [12]	61.00	69.05	55.15	64.20	53.03	67.77	50.03	59.83	42.37	43.56	32.12	41.43	32.28	33.27	28.04	39.62
FedCorr [13]	68.32	60.98	58.82	48.88	73.25	74.59	70.58	75.35	43.96	42.98	42.50	44.78	39.60	51.14	39.66	49.91
FedClean [14]	69.59	65.55	59.93	53.69	70.58	71.33	78.89	79.85	48.88	46.79	47.89	46.55	29.59	40.07	34.19	39.49
FedRG	**72.47**	**76.53**	**74.06**	**77.90**	**83.13**	**87.13**	**83.41**	**88.33**	55.49	53.47	**59.62**	**58.88**	**55.41**	**65.56**	**54.45**	**57.32**

for clean/noisy split, and (iii) EM-style updates using sufficient statistics. These are lightweight compared to the backbone forward/backward. Empirically, the end-to-end wall-clock overhead is approximately $1.32\times$ vs. FedAvg on CIFAR-10/ResNet-18.

4.3. Storage

We report the client-side persistent state beyond transient activations and optimizer states. FedAvg stores only the local model parameters, i.e., $\#\mathrm{Mem}(\mathrm{FedAvg}) = M$. Relative to FedAvg, `FedRG` maintains lightweight geometry-related state in Stage II, including the vMF parameters, the class-to-geometry matrix $\mathbf{B} \in \mathbb{R}^{C\times G}$, the local noise absorption matrix $\mathbf{T}_k \in \mathbb{R}^{C\times C}$, and the running sufficient-statistic buffers used by the online EM updates. Therefore, the extra storage is dominated by

$$\Delta_m = 2Gd + 2CG + C^2 + O(G), \tag{19}$$

where the two Gd terms come from the cluster means and their running vector statistics, the two CG terms come from the class-to-geometry matrix and its accumulated counts, and C^2 is due to the local noise absorption matrix. For CIFAR-10 with $G = 10$, $C = 10$, and $d = 512$, this gives $\Delta_m \approx 10{,}560$ scalars. With a ResNet-18 backbone of $M \approx 11.18$M parameters, the relative overhead is about 0.094%, which remains negligible in practice.

5. Additional Experimental Results

5.1. Experimental Results for SVHN and CIFAR100 under Localized Label Noise

To further validate the robustness of `FedRG` in more realistic federated environments, we additionally report results under the localized label corruption setting on SVHN and CIFAR100 datasets in Tab. 3. In contrast, `FedRG` consistently achieves the best or second-best performance across all evaluated metrics (accuracy, precision, and F1-score) and both symmetric and pairflip localized noise, indicating that the proposed geometry driven noise identification and personalized noise absorption matrix effectively mitigate client-specific label biases. For CIFAR100, we ob-

serve that SymmetricCE is competitive on a subset of metrics in several settings, especially when the noise pattern is more regular and globally shared. However, this advantage is less consistent across the localized settings, whereas FedRG maintains more balanced performance across noise types and evaluation metrics.

5.2. Experimental Results for CIFAR10 with K=100

To examine the scalability of FedRG to larger federations, we further conduct experiments with $K = 100$ clients, while keeping the data partition (Dirichlet with $\alpha = 0.1$) and noise settings identical to those in the main paper. Fig. 2 reports the test accuracy along communication rounds for four label-noise configurations: (a) localized symmetric noise, (b) globalized symmetric noise, (c) localized pairflip noise, and (d) globalized pairflip noise.

Across all four scenarios, increasing the number of clients intensifies statistical heterogeneity and makes optimization more challenging, leading to slower convergence and lower final accuracy for all baselines. Nevertheless, FedRG consistently performs better throughout training, surpassing both classical FL methods and noise-robust approaches.

5.3. Maximum Accuracy Results

According to prior works [13], we additionally report the maximum test accuracy achieved over all communication rounds as a complementary evaluation metric. This provides an upper bound on the performance that each method can reach during training and complements the main results in Sec. 4.3 (main paper), which focus on the overall convergence behavior. We summarize the maximum accuracies on CIFAR10, SVHN, CIFAR100, and CIFAR10* under four label noise settings (LN_1-LN_4) in Tab.4. Across all datasets and noise types, FedRG consistently achieves the best or second-best performance. On CIFAR100, which is more fine-grained and thus more sensitive to label noise, FedRG either matches or surpasses the best competing methods, indicating that our method remains effective in the high-class regime. In the CIFAR10* scenario with 100 clients, FedRG still maintains a substantial advantage, demonstrating that its robustness is preserved when the federation becomes highly heterogeneous and communication is more fragmented. In general, these maximum-accuracy results corroborate the findings of the main article, showing that FedRG not only stabilizes training under severe label noise but also consistently pushes achievable peak performance beyond existing baselines.

5.4. Comparisons with Representation- and Geometry-related Methods

We further compare FedRG with four methods that are closely related to representation learning or geometry-aware modeling, namely FedProto[15], FedCNI[16], GGEUR[17], and FedROM[18]. The experiments are conducted on CIFAR-10 with K = 10 clients under noisy mode **GS** (global symmetric) and **GP** (global pairflip) including clean-label oracle **FedAvg***. The remaining experimental protocol follows the setting used in our main experiments.

Table 5. Comparisons with additional representation- and geometry-related methods on CIFAR10 with $K = 10$ clients. The bold denotes the best result.

	FedAvg*	FedProto	FedCNI	GGEUR	FedROM	FedRG
GS	73.78	45.50	60.44	48.54	60.57	**63.29**
GP		45.22	52.55	41.97	51.67	**64.88**

The results are reported in Tab. 5. Across the two noise settings, FedRG achieves the strongest overall performance among these representation- and geometry-related methods because it does not organize training evidence directly around noisy supervision. Instead, it first learns label-decoupled spherical representations via self-supervision, then models semantic structure with the vMF mixture, and finally identifies noisy samples by comparing label-free geometric evidence with annotated label-conditioned evidence. This design provides a cleaner and more stable criterion for noise identification under heterogeneous federated settings.

5.5. Effect of Shared Self-supervised Representations

We further study whether the gains of FedRG mainly come from stronger feature initialization. To this end, we re-run several representative baselines with the same self-supervised representations and compare them against their original implementations on CIFAR10 with $K = 10$ clients under the globalized symmetric noise setting, while keeping the remaining experimental protocol the same as in the main experiments.

As illustrated in Tab. 6, the results show that sharing the same self-supervised representations does not lead to uniform or consistent improvements: some methods obtain only limited gains, while others even degrade under heterogeneous noisy supervision. This trend becomes even clearer under the more heterogeneous $\alpha = 0.05$ setting, where all compared methods exhibit noticeable performance drops. By contrast, FedRG still achieves the best performance under both the standard setting. This suggests that the advantage of FedRG cannot be explained by representation pretraining alone. Instead, its main benefit comes from the subsequent geometry-guided noise identification procedure, which explicitly models semantic clusters on the hypersphere and measures the agreement between intrinsic geometric evidence and annotated label-conditioned evidence.

Table 6. Effect of shared self-supervised representations. "Origin" denotes the original implementation of each method, "Shared Rep." denotes the variant using the same self-supervised representations, and "$\alpha = 0.05$" denotes a more heterogeneous Dirichlet partition.

Method	Origin	Shared Rep.	$\alpha = 0.05$
SymmetricCE [9]	59.23	61.28	42.99
FedLSR [10]	55.32	48.89	42.12
FedCNI [16]	60.44	61.14	39.24
GGEUR [17]	48.54	44.76	38.98
FedROM [18]	60.57	58.11	43.75
FedRG		**63.29**	**45.52**

Table 7. Hyperparameter sensitivity of FedRG. We report the test accuracy (%) under different hyperparameter choices.

Hypers	Settings	Accuracy (%)
κ	1 / 3 / 5 / 7 / 9	63.28 / 62.88 / 63.29 / 63.07 / 63.33
GMM τ	0.2 / 0.35 / 0.5 / 0.65 / 0.8	61.25 / 62.47 / 63.29 / 63.40 / 62.97
λ_n	0.5 / 0.75 / 1 / 1.25 / 1.5	62.98 / 62.71 / 63.29 / 63.79 / 63.04

5.6. Hyperparameter Sensitivity

We further examine the sensitivity of FedRG to several key hyperparameters. Unless otherwise specified, all settings follow the main experimental protocol. We consider the representative factors, including the initial concentration parameter κ in the vMF model, the threshold τ used in the GMM-based clean/noisy partition, and the loss weight λ_n for the noise absorption term.

The results are reported in Tab. 7. Overall, FedRG is empirically stable across a broad range of hyperparameter choices. For the initialization of the vMF model, varying κ_0 from 1 to 9 leads to only minor fluctuations, which suggests that the subsequent geometry modeling is not overly sensitive to the initial concentration. For the GMM threshold, the best results are obtained around $\tau = 0.5$ and $\tau = 0.65$, while the overall performance remains stable throughout the tested range. For the loss weight λ_n, the accuracy is consistently competitive from 0.5 to 1.5, with the default choice $\lambda_n = 1$ already yielding strong performance. These observations indicate that FedRG does not require over-tuning and remains robust to the main hyperparameters in practice.

6. More Detailed Related Works

Label corruption is especially challenging in FL because supervision is decentralized, heterogeneous, and privacy-constrained [19]. Most federated approaches addressing noisy labels fall into two complementary axes: client-level assessment, which judges which clients are reliable and modulates their influence during training; sample-level assessment, which decides which samples on each device should be trusted or relabeled.

Client-level Assessment. Client-oriented strategies estimate reliability signals from each participant and adapt aggregation or training accordingly. A representative design is the two-phase pipeline of FedNoRo [11]: an unsupervised clustering step (e.g., a GMM [20] over client statistics) partitions clients into presumably clean versus noisy groups, followed by knowledge distillation [21] to steer models from noisy clients toward the global solution. Related two-stage frameworks [12, 22] likewise use client-level reliability to reweight aggregation, freeze or reset unreliable updates, or schedule additional correction steps. These methods share the premise that enhancing client contribution, either through selection or adaptive weighting, can help stabilize global optimization even when per-client supervision is imperfect.

Sample-level Assessment. A parallel literature acts at the sample granularity on each device. FedLSR [10] stabilizes local training through prediction regularization combined with self-distillation [23], reducing overconfident fits to corrupted labels. Two-stage data-quality pipelines [12, 22, 24] remove or relabel suspicious instances before contributing updates. Domain specific work in network traffic [25] adopts the small loss rule to retain examples that train with lower empirical loss, implicitly treating them as cleaner. Some approaches further assume access to a clean auxiliary set to calibrate local decisions or to distill global knowledge [26]; while effective, this assumption is often untenable in privacy-sensitive applications.

Both client-level and sample-level approaches commonly rely on the small loss principle to infer client noise levels or to assess the correctness of individual labels. This reliance can be explicit, for example through the direct thresholding of per-sample losses, or implicit, for instance by using low-loss statistics within GMMs or confidence-based filters. However, deep networks are known to gradually memorize mislabeled data [13], which drives their losses downward and leads to incorrect inclusion of noisy samples in the clean set, creating confirmation bias. At the same time, legitimately difficult but correctly labeled instances, such as minority-class examples or samples near decision boundaries, often retain high loss and risk being mistakenly removed.

To overcome these issues, our work focuses on sample-level noise identification that does not depend on the small loss heuristic and instead exploits the geometric structure of learned representations to achieve more stable and robust noise discrimination in heterogeneous federated settings.

References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017. 1, 6, 7

[2] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In Inderjit S. Dhillon, Dimitris S. Papailiopoulos, and Vivienne Sze, editors, *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*. mlsys.org, 2020. 1, 2, 6, 7

[3] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 10713–10722. Computer Vision Foundation / IEEE, 2021. 1, 2, 6, 7

[4] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34:5972–5984, 2021. 1, 2, 6, 7

[5] Rui Ye, Mingkai Xu, Jianyu Wang, Chenxin Xu, Siheng Chen, and Yanfeng Wang. Feddisco: Federated learning with discrepancy-aware collaboration. In *International Conference on Machine Learning*, pages 39879–39902. PMLR, 2023. 1, 2, 6, 7

[6] Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 18250–18280. PMLR, 2022. 1, 2, 6, 7

[7] Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. Fedexp: Speeding up federated averaging via extrapolation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. 1, 6, 7

[8] Yan Sun, Li Shen, and Dacheng Tao. Understanding how consistency works in federated learning via stage-wise relaxed initialization. *Advances in Neural Information Processing Systems*, 36:80543–80574, 2023. 1, 2, 6, 7

[9] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 322–330. IEEE, 2019. 1, 2, 6, 7, 9

[10] Xuefeng Jiang, Sheng Sun, Yuwei Wang, and Min Liu. Towards federated learning against noisy labels via local self-regularization. In Mohammad Al Hasan and Li Xiong, editors, *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 862–873. ACM, 2022. 1, 2, 6, 7, 9

[11] Nannan Wu, Li Yu, Xuefeng Jiang, Kwang-Ting Cheng, and Zengqiang Yan. Fednoro: Towards noise-robust federated learning by addressing class imbalance and label noise heterogeneity. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 4424–4432. ijcai.org, 2023. 2, 6, 7, 9

[12] Xuefeng Jiang, Sheng Sun, Jia Li, Jingjing Xue, Runhan Li, Zhiyuan Wu, Gang Xu, Yuwei Wang, and Min Liu. Tackling noisy clients in federated learning with end-to-end label correction. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 1015–1026, 2024. 2, 3, 6, 7, 9

[13] Jingyi Xu, Zihan Chen, Tony Q. S. Quek, and Kai Fong Ernest Chong. Fedcorr: Multi-stage federated learning for label noise correction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10174–10183. IEEE, 2022. 2, 3, 6, 7, 8, 9

[14] Xiaoqian Jiang and Jing Zhang. Fedclean: A general robust label noise correction for federated learning. In *Forty-second International Conference on Machine Learning*, 2025. 2, 3, 6, 7

[15] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 8432–8440, 2022. 2, 8

[16] Chenrui Wu, Zexi Li, Fangxin Wang, and Chao Wu. Learning cautiously in federated learning with noisy and heterogeneous clients. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 660–665, 2023. 2, 8, 9

[17] Yanbiao Ma, Wei Dai, Wenke Huang, and Jiayi Chen. Geometric knowledge-guided localized global distribution alignment for federated learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20958–20968, 2025. 2, 8, 9

[18] Xuefeng Jiang, Tian Wen, Sheng Sun, Jinliang Yuan, Huashuo Liu, Peng Li, Lvhua Wu, Yuwei Wang, and Min Liu. Representation optimal matching for federated learning with noisy labels in remote sensing. *IEEE Transactions on Mobile Computing*, 2025. 2, 8, 9

[19] Xuefeng Jiang, Jia Li, Nannan Wu, Zhiyuan Wu, Xujing Li, Sheng Sun, Gang Xu, Yuwei Wang, Qi Li, and Min Liu. Fnbench: Benchmarking robust federated learning against noisy labels. *Authorea Preprints*, 2024. 9

[20] Geoffrey J. McLachlan and Suren I. Rathnayake. On the number of components in a gaussian mixture model. *WIREs Data Mining Knowl. Discov.*, 4(5):341–355, 2014. 9

[21] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 9

[22] Xuefeng Jiang, Peng Li, Sheng Sun, Jia Li, Lvhua Wu, Yuwei Wang, Xiuhua Lu, Xu Ma, and Min Liu. Refining

distributed noisy clients: An end-to-end dual optimization framework. 2025. 9

[23] Sangchul Hahn and Heeyoul Choi. Self-knowledge distillation in natural language processing. In Ruslan Mitkov and Galia Angelova, editors, *Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP 2019, Varna, Bulgaria, September 2-4, 2019*, pages 423–430. INCOMA Ltd., 2019. 9

[24] Xuefeng Jiang, Tian Wen, Zhiqin Yang, Lvhua Wu, Yufeng Chen, Sheng Sun, Yuwei Wang, and Min Liu. Robust federated learning against noisy clients via masked optimization. *arXiv preprint arXiv:2506.02079*, 2025. 9

[25] Siping Shi, Yingya Guo, Dan Wang, Yifei Zhu, and Zhu Han. Distributionally robust federated learning for network traffic classification with noisy labels. *IEEE Transactions on Mobile Computing*, 23(5):6212–6226, 2023. 9

[26] Xiuwen Fang and Mang Ye. Noise-robust federated learning with model heterogeneous clients. *IEEE Transactions on Mobile Computing*, 2024. 9