Efficient Label Distribution Inference Attack and Defense on Classifier Weights in Federated Learning

Daeyoung Choi[1,2] **Gyuejeong Lee**[1*]
[1]Division of AI and Data Science, Korea Cyber University [2]Intellectus
{choidy, lgj413}@koreacu.ac.kr

Abstract

Label distribution leakage in federated learning (FL) poses a critical privacy threat, exposing population-level patterns beyond individual data breaches. Building on known correlations between class sample frequencies and classifier weight magnitudes, we demonstrate how adversaries can exploit these relationships to infer label distributions from shared model parameters in FL. To characterize this threat, we propose WLIA (Weight Norm-based Label Distribution Inference Attack), which infers client label distributions by analyzing variations in classifier weight norms across communication rounds. WLIA operates on the classifier layer without requiring auxiliary data, synthetic samples, or meta-classifiers, enabling broader applicability across heterogeneous model architectures. To counter this threat, we introduce RNR (Randomized Weight Norm Regularization), a defense mechanism that disrupts frequency-weight correlations by randomly regularizing weight norms during local training. RNR adds only a penalty term to the cross-entropy loss, making it efficient and easy to integrate. Comprehensive experiments across six datasets demonstrate that WLIA achieves superior inference accuracy compared to existing attacks, while RNR effectively mitigates leakage with minimal impact on utility.

1 Introduction

Federated learning (FL) enables privacy-preserving collaborative machine learning by distributing the training process across multiple participants without centralizing raw data [28]. Conventional model parameter-sharing FL protocols operate under a fundamental assumption: exchanging model updates is inherently safer than sharing raw data. This privacy guarantee has driven widespread adoption in privacy-critical domains such as healthcare and finance [40, 44, 17, 31]. However, recent research has challenged this foundational premise by demonstrating that model parameters and gradients can leak sensitive information through various privacy attacks, including membership inference [34, 42, 32, 35, 5, 25], model inversion [8, 37, 20, 16, 48], and property inference attacks [9, 30, 38].

In this work, we focus on label distribution inference attacks—a critical category of property inference attacks that target the class distribution of training data—and their defenses. The label distribution, defined as the proportion of samples across classes in a client's local dataset, can reveal sensitive population-level information even when individual samples remain private. For example, inferred label distributions may expose patient demographics in healthcare or risk profiles in finance, constituting serious privacy violations.

Several methods have been proposed to address this vulnerability, including PPA [46], LDIA [10], PGDA [15], SLIA[1] and LDPM [14], and Decaf [6]. These approaches typically assume an honest-

*Corresponding author.

[1]For brevity, we refer to the attack method in [14] as SLIA (Simple Label Distribution Inference Attack). The original authors describe it as a "simple yet effective label distribution inference attack" without using an acronym.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Comparison of label distribution inference attack and defense methods (✓ indicates supported/required; ✗ indicates not supported/not required).

Reference	Capabilities				Attack Requirements		
	Attack	Defense	Label Distribution	Model Heterogeneity	Auxiliary Data	Synthetic Data	Meta Classifier
Zhou *et al.* [46]	PPA	✗	✗	✗	✓	✗	✓
Gu and Bai [10]	LDIA	✗	✓	✓	✓	✗	✓
Jiang *et al.* [15]	PGDA	✗	✓	✗	✗	✓	✗
Jiang *et al.* [14]	SLIA	LDPM	✓	✗	✓	✗	✗
Dai *et al.* [6]	Decaf	✗	✓	✗	✓	✗	✗
Ours	**WLIA**	**RNR**	✓	✓	✗	✗	✗

but-curious server adversary with access to complete model parameters and auxiliary or synthetic datasets. However, they suffer from three critical limitations (Table 1). First, requiring complete model parameters and auxiliary or synthetic datasets imposes impractical computational overhead. Second, the full model requirement restricts these methods to homogeneous FL settings, precluding application to scenarios with heterogeneous client architectures. Third, existing research emphasizes attack methodologies while defensive strategies remain largely underexplored.

To overcome these limitations, we leverage the observation that model parameters encode substantial information about underlying data distributions [18, 19, 23]. Building upon this insight, we propose WLIA (Weight Norm-based Label Distribution Inference Attack), which infers label distributions solely from classifier weight norms by exploiting the strong correlation between ℓ_2-norms of classifier weights and their corresponding class frequencies [18, 23]. Specifically, WLIA computes the norm of weight differences between client and global classifiers for each class, then normalizes these values to estimate the label distribution. To counter WLIA, we propose RNR (Randomized Weight Norm Regularization), which regularizes classifier weights toward randomly sampled targets during local training, breaking the exploitable correlation while preserving model utility.

Together, WLIA and RNR constitute a comprehensive framework for understanding and mitigating label distribution leakage in FL through weight-norm analysis. Both methods operate exclusively on classifier weights, enabling compatibility with heterogeneous architectures where clients share only output layer parameters. Our contributions are summarized as follows (Table 1):

- We propose WLIA, a weight-based label inference attack that recovers client label distributions directly from classifier weights without auxiliary data, synthetic data generation, or meta-classifier training—operating under minimal assumptions about the adversary.
- We introduce RNR, a defense mechanism that mitigates WLIA by decoupling weight norms from class frequencies via randomized norm regularization, achieving substantial privacy protection with negligible computational overhead.
- Through extensive evaluation across six datasets and diverse federated settings, we demonstrate that WLIA achieves superior inference accuracy over existing attacks. At the same time, RNR effectively counters WLIA with minimal utility degradation.

2 Related Work

Label Distribution Inference in FL. Recent work has revealed that federated learning is vulnerable to label distribution inference attacks that recover class proportions in clients' private data. PPA [46] exploits gradient sensitivity to class sample proportions to infer top-k preference distributions. LDIA [10] trains a neural network to characterize output layer updates for inferring label distributions. PGDA [15] employs projected gradient descent to infer both feature and label distributions of each party's training data. Jiang *et al.* [14] propose a simple label distribution inference attack (which we call SLIA) and provide formal privacy guarantees for label distribution protection via SGD perturbation (LDPM). Decaf [6] decomposes clients' data distributions by analyzing local gradient magnitudes. However, these attacks typically assume a complete model, rendering them inapplicable to classifier-sharing heterogeneous FL, where only classifier parameters are transmitted.

Classifier-Sharing in Heterogeneous FL. Heterogeneous FL represents a paradigm that allows clients to utilize diverse model architectures while handling heterogeneous data distributions [41].

Existing heterogeneous FL methods differ in the type of information exchanged between the server and clients, ranging from model parameters [26, 12] and auxiliary models [33, 43, 27] to data generator parameters [47] and logits or representations [13, 36, 45, 24]. Our work focuses on heterogeneous FL paradigms in which clients transmit classifiers while preserving feature extractors locally. This configuration deviates from scenarios where models are shared, making conventional label distribution inference attacks impossible. Notable approaches include LG-FedAvg [26], which shares upper layers while preserving client-specific lower layers, and FedClassAvg [12], which aggregates classifier weights to align decision boundaries and improve performance on underrepresented classes under non-independent and identically distributed (non-IID) data. CHFL [27] introduces a framework leveraging parameter decoupling and data-free knowledge distillation via classifier-only communication.

3 Background

We provide background on federated learning and label distribution inference from weight norms.

3.1 Federated Learning Framework

Consider a federated learning system with M clients collaboratively training deep networks for K-class classification under server coordination. Each client i maintains a local dataset $\mathcal{D}_i$ drawn from distribution P_i. Each client's model $\boldsymbol{w}_i$ comprises a feature extractor f_i parameterized by $\boldsymbol{\theta}_i \in \mathbb{R}^d$ that maps inputs from $\mathbb{R}^D$ to a d-dimensional representation, and a classifier parameterized by $\boldsymbol{\phi}_i \in \mathbb{R}^{K\times d}$ that produces class predictions. Let $\ell(\boldsymbol{\theta}_i, \boldsymbol{\phi}_i; x, y)$ denote the loss function measuring classification performance on client i's local distribution P_i. The federated learning objective minimizes the aggregate expected loss:

$$\min_{\mathbf{W}} \left\{ F(\mathbf{W}) := \frac{1}{M} \sum_{i=1}^{M} \mathbb{E}_{(x,y)\sim P_i} \left[\ell(\boldsymbol{\theta}_i, \boldsymbol{\phi}_i; x, y) \right] \right\}, \tag{1}$$

where $\mathbf{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M]$ denotes all client models. In practice, we approximate this via empirical risk minimization over local data $\mathcal{D}_i = \{(x_i^{(l)}, y_i^{(l)})\}_{l=1}^{n_i}$:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}_i(\boldsymbol{\theta}_i, \boldsymbol{\phi}_i), \tag{2}$$

where $\mathcal{L}_i(\boldsymbol{\theta}_i, \boldsymbol{\phi}_i) = \frac{1}{n_i} \sum_{l=1}^{n_i} \ell(\boldsymbol{\theta}_i, \boldsymbol{\phi}_i; x_i^{(l)}, y_i^{(l)})$ is the empirical loss for client i.

3.2 Weight Norms as Label Distribution Indicators

Prior work has established fundamental connections between training dynamics and label distribution in deep neural networks. Anand *et al.* [2] proved that under class imbalance, the expected squared gradient magnitude exhibits a positive correlation with class sample size, yielding proportionally larger gradients for majority classes. Building on this theoretical foundation, Kang *et al.* [18] and Lee and Choi [23] demonstrated that classifier weight norms scale with the number of training samples per class. More specifically, Lee and Choi [23] established a positive correlation between weight norm $\|\boldsymbol{\phi}_{i,j}\|$ and class sample count $n_{i,j}$. Here, $\boldsymbol{\phi}_{i,j} \in \mathbb{R}^d$ denotes the j-th row of client i's classifier weight matrix $\boldsymbol{\phi}_i$, which connects the penultimate layer to the output neuron for class j, and $n_{i,j}$ represents the number of class-j samples in client i's local dataset.

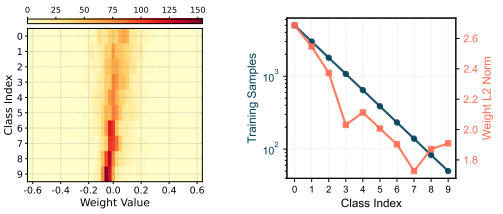
Following Lee and Choi [23], we denote client i's true label distribution as $\mathbf{p}_i = (p_{i,1}, \ldots, p_{i,K})$, where $p_{i,j}$ represents the proportion of class-j samples. The empirical label distribution $\hat{\mathbf{p}}_i = (\hat{p}_{i,1}, \ldots, \hat{p}_{i,K})$ is given by:

$$\hat{p}_{i,j} = \frac{n_{i,j}}{\sum_{k=1}^{K} n_{i,k}} = \frac{n_{i,j}}{n_i}, \tag{3}$$

where $n_i = |\mathcal{D}_i|$ denotes the total number of samples in client i's dataset. We then define the weight norm-based label distribution estimate $\mathbf{q}_i = (q_{i,1}, \ldots, q_{i,K})$, where each component is computed as:

$$q_{i,j} = \frac{\|\boldsymbol{\phi}_{i,j}\|_2}{\sum_{k=1}^{K} \|\boldsymbol{\phi}_{i,k}\|_2}. \tag{4}$$

Key Observation. We observe that $\mathbf{q}_i$ can serve as an effective proxy for underlying label distributions. To validate this relationship, we trained ResNet-18 on a long-tailed CIFAR-10 with a single client i, where class frequencies follow an exponential decay with an imbalance ratio of 100:1 (from 5,000 to 50 samples per class).



(a) Weight distribution (b) Inferred Distribution

Figure 1: Output layer weight and sample distributions on long-tailed CIFAR-10 **without RNR**.

We visualized the output layer weight distributions for each class as a heatmap (Figure 1a). Each row corresponds to the d weight values in $\boldsymbol{\phi}_{i,j} \in \mathbb{R}^d$, where color intensity indicates the frequency of weights within each bin. The heatmap reveals a clear pattern: classes with more training samples exhibit weight distributions with substantially higher magnitudes and greater spread, while minority classes display concentrated distributions near zero. This phenomenon arises because frequent classes receive more gradient updates during training, leading to an accumulation of weight magnitudes. Consequently, weight norm distributions (red) strongly correlate with training sample distributions (navy), demonstrating that adversaries can infer label distributions from output layer weights alone (Figure 1b). Results on CIFAR-100 exhibit similar patterns (Appendix B).

Challenges for Federated Learning. However, weight norm-based inference using Eq. (4) suffers from limited precision, particularly for tail classes with few samples (Figure 1b). Moreover, and critically, whether these observations extend to FL remains an open question, as parameter aggregation across clients with heterogeneous local distributions may disrupt the weight norm-sample frequency relationship. To address these challenges, we develop a principled attack framework that refines weight-based inference and systematically evaluates its effectiveness in federated learning settings.

4 Methodology

We first formalize the threat model, then present our attack and defense mechanisms.

4.1 Threat Model and Privacy Metric

Threat Model. We consider an honest-but-curious server that faithfully executes the federated learning protocol while attempting to infer clients' private label distributions. Consistent with prior label inference attacks such as PPA, LDIA, PGDA, SLIA, and Decaf, the server observes the plaintext model weights $\boldsymbol{w}_i^{(t)}$ uploaded by each client at each communication round t, has access to the previous global model $\boldsymbol{w}_g^{(t-1)}$, and knows the model architecture; however, it cannot access raw data, gradients, or auxiliary datasets. Our attack targets only the classifier layer weights $\boldsymbol{\phi}_i^{(t)}$, aligning with heterogeneous FL (HtFL) settings where only classifier parameters are shared to support heterogeneous architectures or reduce communication overhead [26, 12, 27].

WLIA therefore does not apply in two cases. First, it cannot be used when secure aggregation, homomorphic encryption, or secure multi-party computation conceals per-client updates; such protections remain relatively uncommon in large-scale deployments because of their computational overhead, the complexity of handling client dropout, and their vulnerability to collusion [4, 46, 6, 15, 14]. Second, among HtFL methods, it does not apply to approaches that exchange logits, representations, or other non-classifier information. We discuss the generality and practicality of our threat model in Appendix A.

Privacy Metric. We formalize a privacy metric to quantify the leakage of client label distributions. Specifically, we measure leakage with respect to the empirical label distribution $\hat{\mathbf{p}}_i$ from client i's finite training dataset, which represents the practical privacy target.

Definition 1 (Label Distribution Leakage) *Let $\hat{\mathbf{p}}_i$ denote client i's empirical label distribution and $\tilde{\mathbf{p}}_i$ denote the adversary's estimated distribution. The* label distribution leakage *is:*

$$\mathrm{LDL}_i(\tilde{\mathbf{p}}_i) := D(\hat{\mathbf{p}}_i \| \tilde{\mathbf{p}}_i), \tag{5}$$

where $D(\cdot\|\cdot)$ is a statistical divergence measure. Lower LDL_i values indicate more accurate estimation and thus higher privacy leakage.

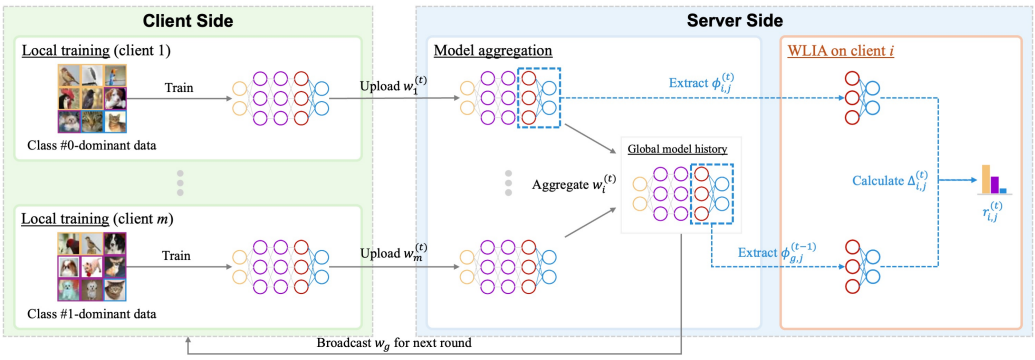


Figure 2: Overview of the Federated Averaging (FedAvg) protocol and the WLIA mechanism.

We employ two complementary metrics that capture different aspects of privacy risk. Total Variation Distance (TV) treats all classes equally, whereas Kullback-Leibler Divergence (KL) weights errors by the empirical distribution and thus emphasizes discrepancies in high-probability classes:

$$\mathrm{LDL}_i^{\mathrm{TV}}(\tilde{\mathbf{p}}_i) = \frac{1}{2}\sum_{j=1}^{K} |\hat{p}_{i,j} - \tilde{p}_{i,j}|, \qquad \mathrm{LDL}_i^{\mathrm{KL}}(\tilde{\mathbf{p}}_i) = \sum_{j=1}^{K} \hat{p}_{i,j} \log \frac{\hat{p}_{i,j}}{\tilde{p}_{i,j}}. \tag{6}$$

To assess system-level privacy, we report the average leakage over all clients:

$$\overline{\mathrm{LDL}} = \frac{1}{M}\sum_{i=1}^{M} \mathrm{LDL}_i. \tag{7}$$

4.2 Attack Method: Weight Norm-Based Label Distribution Inference Attack (WLIA)

In centralized learning, weight norm-based estimates strongly correlate with empirical label distributions (Figure 1b). We adapt this to federated learning by computing norms of inter-round weight updates rather than absolute magnitudes, exploiting FL's iterative communication to enhance inference accuracy.

WLIA functions as a passive inference attack within standard aggregation-based FL protocols such as FedAvg. At each round t, the adversary computes the ℓ_2-norm of the weight difference between client i's classifier and the previous global classifier for each class j (Figure 2):

$$\Delta_{i,j}^{(t)} = \|\boldsymbol{\phi}_{i,j}^{(t)} - \boldsymbol{\phi}_{g,j}^{(t-1)}\|_2, \tag{8}$$

where $\boldsymbol{\phi}_{i,j}^{(t)}$ denotes client i's classifier weights for class j at round t, and $\boldsymbol{\phi}_{g,j}^{(t-1)}$ represents the corresponding global weights from round $t-1$. The key insight is that classes with more training samples lead to larger deviations in the weight parameters from the global model during local optimization.

To derive a probability distribution, we normalize the weight difference norms across K classes. The estimated label distribution $\mathbf{r}_i^{(t)} = (r_{i,1}^{(t)}, \ldots, r_{i,K}^{(t)})$ for client i at t is:

$$r_{i,j}^{(t)} = \frac{\Delta_{i,j}^{(t)}}{\sum_{k=1}^{K} \Delta_{i,k}^{(t)}}. \tag{9}$$

4.3 Defense Method: Randomized Weight Norm Regularization (RNR)

RNR decouples the correlation between $\mathbf{r}_i$ exploited by WLIA and the empirical label distribution $\hat{\mathbf{p}}_i$ by regularizing classifier weight norms toward random targets during local training. Specifically, before the first local training begins, each client i independently samples random target norms $\{\tau_{i,j}\}_{j=1}^{K}$ from $\mathcal{N}(\mu, \sigma^2)$ for each class j. The RNR loss term is:

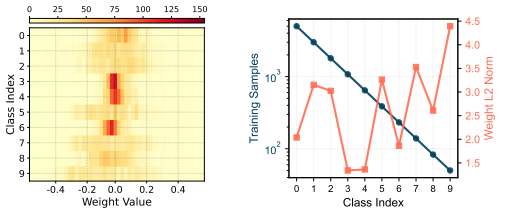
$$\mathcal{L}_{\mathrm{RNR}}(\boldsymbol{\phi}_i) = \sum_{j=1}^{K} \left(\|\boldsymbol{\phi}_{i,j}\|_2 - \tau_{i,j}\right)^2. \tag{10}$$

Combining this with the standard cross-entropy loss yields the total objective:

$$\mathcal{L}_i(\boldsymbol{w}_i) = \mathcal{L}_{\text{CE}}(\boldsymbol{w}_i) + \lambda \mathcal{L}_{\text{RNR}}(\boldsymbol{\phi}_i), \tag{11}$$

where λ controls the privacy-utility trade-off. The sampled targets $\{\tau_{i,j}\}_{j=1}^{K}$ remain fixed across all rounds. Crucially, even if the adversary knows these targets, WLIA becomes ineffective since the observed weight norms now reflect both the label distribution and the random targets, which have no systematic relationship with true class frequencies.

To validate RNR, we applied it to the example in Section 3.2 with random targets sampled from $\mathcal{N}(\mu = 3, \sigma^2 = 1)$. Figure 3a shows that classifier weights now exhibit varied distributions across classes regardless of sample size, unlike the undefended case in Figure 1a. Figure 3b shows that weight norms no longer correlate with the empirical label distribution and instead follow the random targets. CIFAR-100 exhibits similar results (Figure 10 in Appendix B). Algorithm 1 presents WLIA and RNR within the FedAvg framework.



(a) Weight distribution (b) Inferred Distribution

Figure 3: Output layer weights and sample distributions on long-tailed CIFAR-10 **with RNR**.

Algorithm 1 WLIA Attack and RNR Defense in FedAvg

Input: M clients, K classes, T communication rounds
Hyperparameters: Regularization strength λ, target norm parameters μ, σ^2
Output: Inferred label distributions $\{\mathbf{r}_i^{(T)}\}_{i=1}^{M}$ for all clients
Server executes:

Initialize global model parameters $\boldsymbol{w}_g^{(0)}$
Initialize $\mathbf{r}_i^{(0)} \leftarrow \text{Uniform}(K)$ for all clients $i \in [M]$
for communication round $t = 1, \ldots, T$ **do**
  Sample client subset $\mathcal{C}^{(t)} \subseteq [M]$
  for each client $i \in \mathcal{C}^{(t)}$ **in parallel do**
    $\boldsymbol{w}_i^{(t)} \leftarrow \text{ClientUpdate}(i, \boldsymbol{w}_g^{(t-1)}, t)$
  for each client $i \in \mathcal{C}^{(t)}$ **do** ▷ **Attack: Label Distribution Inference with WLIA**
    for class $j = 1, \ldots, K$ **do**
      Compute $\Delta_{i,j}^{(t)} = \|\boldsymbol{\phi}_{i,j}^{(t)} - \boldsymbol{\phi}_{g,j}^{(t-1)}\|_2$ ▷ Eq. (8)
    Infer $r_{i,j}^{(t)} = \frac{\Delta_{i,j}^{(t)}}{\sum_{k=1}^{K} \Delta_{i,k}^{(t)}}$ for all $j \in [K]$ ▷ Eq. (9)
  Aggregate $\boldsymbol{w}_g^{(t)} = \frac{1}{|\mathcal{C}^{(t)}|} \sum_{i \in \mathcal{C}^{(t)}} \boldsymbol{w}_i^{(t)}$
return $\{\mathbf{r}_i^{(T)}\}_{i \in [M]}$

Procedure $\text{ClientUpdate}(i, \boldsymbol{w}_g^{(t-1)}, t)$**:**

Initialize $\boldsymbol{w}_i^{(t)} \leftarrow \boldsymbol{w}_g^{(t-1)}$
if $t = 1$ **then**
  for class $j = 1, \ldots, K$ **do**
    Sample $\tau_{i,j} \sim \mathcal{N}(\mu, \sigma^2)$ and store for all rounds
for local epoch $e = 1, \ldots, E$ **do** ▷ **Defense: Local Training with RNR**
  for each mini-batch $b \in \mathcal{B}_i$ **do**
    Compute $\mathcal{L}_{\text{CE}} = \frac{1}{|b|} \sum_{(x,y) \in b} \ell(f(\boldsymbol{w}_i; x), y)$
    Compute $\mathcal{L}_{\text{RNR}} = \sum_{j=1}^{K} \left(\|\boldsymbol{\phi}_{i,j}\|_2 - \tau_{i,j}\right)^2$ ▷ Eq. (10)
    Compute $\mathcal{L}_i = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{RNR}}$ ▷ Eq. (11)
    Update $\boldsymbol{w}_i^{(t)} \leftarrow \boldsymbol{w}_i^{(t)} - \eta \nabla_{\boldsymbol{w}} \mathcal{L}_i$
return $\boldsymbol{w}_i^{(t)}$

5 Experiments

This section presents our experimental setup and results. Our code is available at `https://github.com/lgj413-koreacu/WLIA-RNR`.

5.1 Experimental Setup

Datasets and Models. We evaluated our approach across diverse data modalities: Driver Diagnosis (Drive) for tabular data, DBpedia [3] for text classification, and MNIST [7], Fashion-MNIST (FMNIST) [39], CIFAR-10, and CIFAR-100 [22] for image classification. Dataset characteristics are summarized in Table 6 (Appendix C). Model architectures were selected based on task modality and complexity. For Drive, we employed a 2-layer multilayer perceptron (MLP); for text classification, we used TextCNN [21]. For image tasks, we employed a 4-layer convolutional neural network (CNN) on MNIST and FMNIST, and ResNet-18 [11] (default unless otherwise specified) and MobileViT [29] on CIFAR-10/100.

Baselines and Evaluation Protocol. Among the methods in Table 1, we compared WLIA against four reproducible baselines: PGDA, SLIA, Decaf, and random guess, all of which provide publicly available code or detailed implementation specifications. A random guess serves as a lower bound on attack performance, generating valid distributions by sampling from a uniform distribution $\mathcal{U}(0, 1)$ and normalizing to satisfy the probability simplex constraint, following [6]. For auxiliary data-based methods (SLIA and Decaf), we adopted the experimental settings from [6], utilizing 10 auxiliary samples per class to approximate the global data distribution. Following standard practice in FL research [45], we computed the averaged label distribution $\overline{\text{LDL}}$ and average client accuracy at every communication round, reporting the lowest $\overline{\text{LDL}}$ as the final performance metric. We repeated each experiment three times with different random seeds and reported the average results for statistical reliability.

Federated Learning Setup. We configured a federated system with 10 clients in full participation mode. Training was conducted over 20 rounds with a learning rate of 0.005 (0.05 for TextCNN), batch size of 10, and 10 local epochs per client. Data heterogeneity was simulated using a Dirichlet distribution with concentration parameter $\alpha = 0.5$ to create non-IID partitions. We evaluated two categories of FL algorithms: (1) a *homogeneous method* that shares the full model (FedAvg), and (2) *heterogeneous methods* that share only the output layer (LG-FedAvg, FedClassAvg, and CHFL). For RNR, random targets were sampled with $\mu = 3.0$ and $\sigma = 1.0$, using a regularization parameter $\lambda = 0.1$ for DBpedia, MNIST, and FMNIST, and $\lambda = 1.0$ for the remaining datasets. Implementation details are provided in Appendix C.

5.2 Attack Performance and Analysis

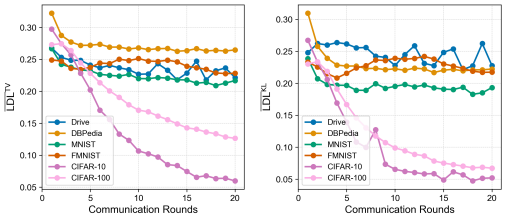
Overall Performance. Table 2 presents label distribution inference accuracy across six datasets. WLIA consistently outperforms all baselines, achieving the lowest $\overline{\text{LDL}}^{\text{TV}}$ and $\overline{\text{LDL}}^{\text{KL}}$ across all settings. Notably, WLIA maintains superior performance even on challenging datasets such as CIFAR-100, where baseline methods struggle due to increased complexity. Regarding PGDA, which generates synthetic data from model parameters, the original work only applies to low-dimensional datasets and does not consider image datasets. We observed that PGDA generally struggles with high-dimensional image datasets, including MNIST. CIFAR-10/100 results on MobileViT are in Table 9 (Appendix D.2).

Table 2: Comparison of label distribution leakage (TV: $\overline{\text{LDL}}^{\text{TV}}$, KL: $\overline{\text{LDL}}^{\text{KL}}$) across six datasets. Complete results with standard deviations are in Tables 7 and 8 (Appendix D.1).

Attack	Drive		DBpedia		MNIST		FMNIST		CIFAR-10		CIFAR-100	
	TV	KL	TV	KL	TV	KL	TV	KL	TV	KL	TV	KL
Random	0.455	0.638	0.453	0.615	0.451	0.600	0.528	0.817	0.452	0.614	0.482	0.709
PGDA	0.304	0.355	0.278	0.277	0.589	1.442	0.575	1.329	0.381	0.603	0.444	0.709
SLIA	0.222	0.254	0.269	0.442	0.315	1.275	0.284	0.517	0.215	0.188	0.197	0.127
Decaf	0.312	0.612	0.423	0.535	0.405	0.538	0.368	0.617	0.182	1.779	0.391	0.587
WLIA	**0.203**	**0.189**	**0.242**	**0.237**	**0.202**	**0.149**	**0.279**	**0.252**	**0.052**	**0.028**	**0.129**	**0.061**

Table 3: $\overline{\text{LDL}}^{\text{TV}}$ for varying α and M on CIFAR-100.

Attack	Data Heterogeneity (α)		Num. of Clients (M)	
	$\alpha = 0.1$	$\alpha = 1.0$	$M = 50$	$M = 100$
Random	0.718	0.376	0.512	0.504
PGDA	0.479	0.441	0.424	0.405
SLIA	0.189	0.203	0.318	0.312
Decaf	0.558	0.329	0.442	0.449
WLIA	**0.185**	**0.108**	**0.156**	**0.155**



(a) $\overline{\text{LDL}}^{\text{TV}}$ (b) $\overline{\text{LDL}}^{\text{KL}}$

Figure 4: WLIA performance across FL rounds.

Table 4: $\overline{\text{LDL}}^{\text{TV}}$ results under heterogeneous FL settings using WLIA, the only viable inference method, as baselines (PGDA, SLIA, Decaf) require full global model parameters unavailable in HtFL.

FL method	DBpedia	FMNIST	CIFAR-10	CIFAR-100
LG-FedAvg	0.276	0.323	0.055	0.142
FedClassAvg	0.351	0.345	0.367	0.189
CHFL	0.415	0.305	0.195	0.320

Table 5: Memory requirements and per-round execution time for WLIA and baselines, where WLIA stores only classifier parameters $|\boldsymbol{\phi}|$ with faster execution.

Attack	Memory Requirement	Execution Time (s)	
		Drive	CIFAR-10
PGDA	$(M+1) \times (\lvert\boldsymbol{\theta}\rvert + \lvert\boldsymbol{\phi}\rvert) \times t$	35.964	284.974
SLIA	$(M+1) \times (\lvert\boldsymbol{\theta}\rvert + \lvert\boldsymbol{\phi}\rvert) \times t$	0.262	6.148
Decaf	$(M+1) \times (\lvert\boldsymbol{\theta}\rvert + \lvert\boldsymbol{\phi}\rvert) \times t$	1.925	12.023
WLIA	$(M+1) \times \lvert\boldsymbol{\phi}\rvert$	**3.13e-4**	**0.001**

Data Heterogeneity and Client Scalability. Label distribution inference becomes challenging as data heterogeneity and the number of clients increase, as these factors amplify local distribution diversity. To evaluate robustness in such settings, we varied the heterogeneity level (α) and client count (M). Table 3 shows that WLIA consistently achieves superior performance across all configurations on CIFAR-100. Generally similar trends are observed for other datasets in Tables 10 and 11 (Appendix D.3).

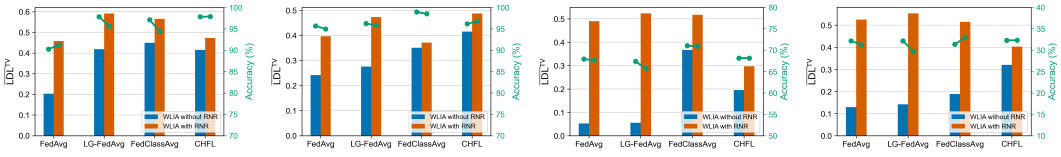
Performance on Heterogeneous FL. Table 4 presents only WLIA results across three HtFL methods, demonstrating practical label distribution inference with accuracy comparable to homogeneous FedAvg (Table 2). Notably, PGDA, SLIA, and Decaf are inapplicable to HtFL settings because they rely on complete model parameters, which are not typically exchanged in heterogeneous architectures. Thus, WLIA is the only tested method that is viable for label distribution inference in HtFL.

Convergence Analysis. We analyze WLIA's performance across communication rounds. Figure 4 shows $\overline{\text{LDL}}^{\text{TV}}$ and $\overline{\text{LDL}}^{\text{KL}}$ across six datasets over 20 rounds. Both metrics exhibit a downward trend as training progresses, indicating improved inference accuracy over time.

Computational Efficiency. Table 5 compares memory requirements and per-round execution time on Drive and CIFAR-10, showing WLIA's superior efficiency. WLIA maintains constant memory complexity by requiring only the current and previous round parameters, whereas PGDA and SLIA store all historical model updates, resulting in memory that scales linearly with round t. WLIA also achieves significantly lower execution time by computing a single ℓ_2 distance between consecutive classifier weights per round. In contrast, baseline methods require multiple forward passes through the whole model using auxiliary or synthetic data, resulting in substantial computational overhead.

5.3 Defense Performance and Analysis

Overall Performance. Figure 5 illustrates the defense effectiveness of RNR against WLIA under both FedAvg and HtFL methods. RNR consistently degrades WLIA's attack performance across all configurations, as indicated by the substantial increase in $\overline{\text{LDL}}^{\text{TV}}$ (blue to red bars), while maintaining minimal accuracy degradation (green lines).



(a) Drive (b) DBpedia (c) CIFAR-10 (d) CIFAR-100

Figure 5: Defense performance of RNR against WLIA on FedAvg and HtFL methods.

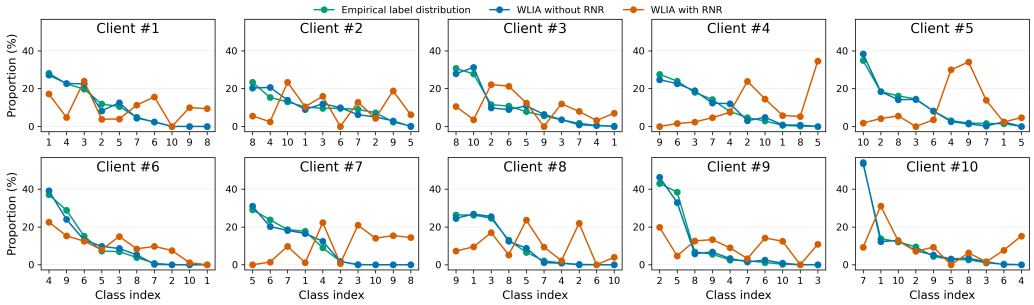
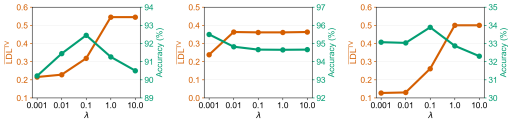
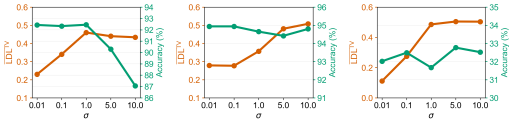


Figure 6: Client-level defense performance of RNR on CIFAR-10 against WLIA under FedAvg.



(a) Drive (b) DBpedia (c) CIFAR-100

Figure 7: Impact of λ on defense performance and accuracy under FedAvg with RNR.



(a) Drive (b) DBpedia (c) CIFAR-100

Figure 8: Impact of σ on defense performance and accuracy under FedAvg with RNR.

Per-Client Performance. Figure 6 compares empirical and inferred label distributions for individual CIFAR-10 clients under FedAvg, with classes ranked by decreasing training data proportion. WLIA without RNR (blue, $\mathbf{r}_i$) accurately infers the ground-truth (green, $\hat{\mathbf{p}}_i$), demonstrating its effectiveness. With RNR (red), WLIA's inferences deviate substantially from both the ground truth and the undefended baseline, confirming effective defense against label distribution inference.

Hyperparameter Sensitivity Analysis. Two hyperparameters govern RNR's performance: the regularization strength λ and target norm separation σ. Figures 7 and 8 demonstrate that larger values of both λ and σ improve defense performance (higher $\overline{\text{LDL}}^{\text{TV}}$), with σ providing enhanced obfuscation through increased inter-class norm separation. Importantly, accuracy exhibits no systematic dependence on these hyperparameters within the explored ranges. Notably, μ serves merely as a reference for the target norm with negligible impact on defense performance, whereas σ governs inter-class norm separation and is the primary driver of privacy protection. We therefore focus our sensitivity analysis on σ and λ.

Impact of RNR on Convergence. A natural concern is whether the regularization term in RNR disrupts convergence by biasing local updates away from the empirical risk minimizer. We address this in Appendix E.3, where we show empirically that RNR preserves global convergence and local training stability.

Comparison to Differential Privacy. We compare RNR with differential privacy (DP) for defending against inference of label distributions. RNR achieves comparable privacy protection with better model utility (Appendix E.4).

Complete experimental results across all datasets and settings are provided in Appendix E.

6 Limitations and Conclusion

Our approaches have three limitations. The first two concern the conditions under which WLIA applies. First, WLIA requires access to classifier parameters and therefore does not apply to HtFL methods in which clients share only other information, such as logits or intermediate representations. Second, WLIA requires the server to observe individual, unencrypted classifier updates. It is therefore not applicable under secure aggregation or other cryptographic defenses that conceal individual updates, and we leave a systematic study under such protections to future work. Differential privacy, in contrast, does not conceal updates, and it reduces WLIA's accuracy only at privacy budgets that also severely degrade model utility. Third, RNR targets attacks that exploit weight-norm patterns and offers limited protection against other attack mechanisms (Appendix E.5).

In this work, we introduced WLIA and RNR to address label distribution leakage in FL. WLIA achieves superior attack performance to existing methods without requiring auxiliary data or meta-classifiers, while also supporting heterogeneous architectures. RNR provides effective defense with minimal overhead and preserves model accuracy. RNR offers a favorable trade-off between defense and utility, providing a practical, lightweight solution for privacy-preserving federated learning.

Acknowledgments and Disclosure of Funding

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (No. RS-2026-25487032, Research on Secure and Resource-Efficient Federated Learning Framework for Defense AI; No. RS-2026-25595794, Context-Conditioned Weight Steering Vector Composition for Controllable Alignment of On-device Language Models) and by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2023-00235293, Development of autonomous driving big data processing, management, search, and sharing interface technology to provide autonomous driving data according to the purpose of usage).

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Rangachari Anand, Kishan G Mehrotra, Chilukuri K Mohan, and Sanjay Ranka. An improved algorithm for neural network classification of imbalanced training sets. *IEEE transactions on neural networks*, 4(6):962–969, 1993.

[3] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *international semantic web conference*, pages 722–735. Springer, 2007.

[4] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[5] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974. PMLR, 2021.

[6] Zhiyang Dai, Yansong Gao, Chunyi Zhou, Anmin Fu, Zhi Zhang, Minhui Xue, Yifeng Zheng, and Yuqing Zhang. Decaf: Data distribution decompose attack against federated learning. *IEEE Transactions on Information Forensics and Security*, 2024.

[7] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

[8] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[9] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 619–633, 2018.

[10] Yuhao Gu and Yuebin Bai. Ldia: Label distribution inference attack against federated learning in edge computing. *Journal of Information Security and Applications*, 74:103475, 2023.

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

[12] Jaehee Jang, Heoneok Ha, Dahuin Jung, and Sungroh Yoon. Fedclassavg: Local representation learning for personalized federated learning on heterogeneous neural networks. In *Proceedings of the 51st international conference on parallel processing*, pages 1–10, 2022.

[13] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

[14] Yangfan Jiang, Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, and Beng Chin Ooi. Protecting label distribution in cross-silo federated learning. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 4828–4847. IEEE, 2024.

[15] Yangfan Jiang, Xinjian Luo, Yuncheng Wu, Xiaochen Zhu, Xiaokui Xiao, and Beng Chin Ooi. On data distribution leakage in cross-silo federated learning. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3312–3328, 2024.

[16] Mostafa Kahla, Si Chen, Hoang Anh Just, and Ruoxi Jia. Label-only model inversion attacks via boundary repulsion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15045–15053, 2022.

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

[18] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations (ICLR)*, 2020.

[19] Ashkan Khakzar, Soroosh Baselizadeh, Saurabh Khanduja, Christian Rupprecht, Seong Tae Kim, and Nassir Navab. Neural response interpretation through the lens of critical pathways. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13528–13538, 2021.

[20] Mahdi Khosravy, Kazuaki Nakamura, Yuki Hirose, Naoko Nitta, and Noboru Babaguchi. Model inversion attack by integration of deep generative models: Privacy-sensitive face generation from a face recognition system. *IEEE Transactions on Information Forensics and Security*, 17:357–372, 2022.

[21] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[23] Gyuejeong Lee and Daeyoung Choi. Class-wise federated averaging for efficient personalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1773–1782, 2025.

[24] Gyuejeong Lee and Daeyoung Choi. Communication-efficient heterogeneous federated learning with sparse prototypes in resource-constrained environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 22787–22795, 2026.

[25] Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 880–895, 2021.

[26] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

[27] Chang Liu, Yuwen Yang, Xun Cai, Yue Ding, and Hongtao Lu. Completely heterogeneous federated learning. *arXiv preprint arXiv:2210.15865*, 2022.

[28] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[29] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021.

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

[31] Sarthak Pati, Sourav Kumar, Amokh Varma, Brandon Edwards, Charles Lu, Liangqiong Qu, Justin J Wang, Anantharaman Lakshminarayanan, Shih-han Wang, Micah J Sheller, et al. Privacy preservation for federated learning in health care. *Patterns*, 5(7), 2024.

[32] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.

[33] Tao Shen, Jie Zhang, Xinkang Jia, Fengda Zhang, Gang Huang, Pan Zhou, Kun Kuang, Fei Wu, and Chao Wu. Federated mutual learning. *arXiv preprint arXiv:2006.16765*, 2020.

[34] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[35] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX security symposium (USENIX security 21)*, pages 2615–2632, 2021.

[36] 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.

[37] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pages 601–618, 2016.

[38] Xiuling Wang and Wendy Hui Wang. Group property inference attacks against graph neural networks. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2871–2884, 2022.

[39] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[40] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[41] Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.

[42] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[43] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328, 2018.

[44] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[45] Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16768–16776, 2024.

[46] Chunyi Zhou, Yansong Gao, Anmin Fu, Kai Chen, Zhiyang Dai, Zhi Zhang, Minhui Xue, and Yuqing Zhang. Ppa: Preference profiling attack against federated learning. *arXiv preprint arXiv:2202.04856*, 2022.

[47] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International conference on machine learning*, pages 12878–12889. PMLR, 2021.

[48] Tianqing Zhu, Dayong Ye, Shuai Zhou, Bo Liu, and Wanlei Zhou. Label-only model inversion attacks: Attack with the least information. *IEEE Transactions on Information Forensics and Security*, 18:991–1005, 2022.

Appendix

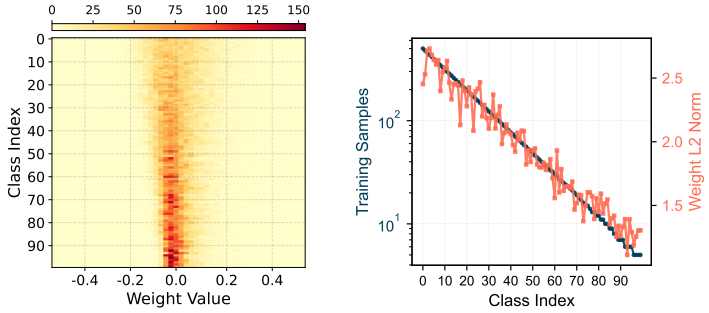
A Details on Threat Model

Generality. Our attack is architecture-agnostic and task-agnostic, requiring only a linear classifier layer. It exploits the fundamental relationship between classifier weight norms and the frequencies of training samples. This property emerges from gradient-based training dynamics, independent of the feature extractor (MLPs, CNNs, RNNs, transformers) or the task domain. We demonstrate this broad applicability across diverse architectures and tasks in Section 5.

Practicality. Our attack offers significant practical advantages over existing inference attacks. Unlike gradient-based attacks requiring iterative optimization with auxiliary datasets, WLIA operates through direct observation with $\mathcal{O}(Kd)$ computational complexity—merely computing ℓ_2-norms of classifier weights (a small fraction of total parameters). This eliminates the need for auxiliary data, iterative procedures, or specialized infrastructure, making WLIA immediately deployable by real-world adversaries.

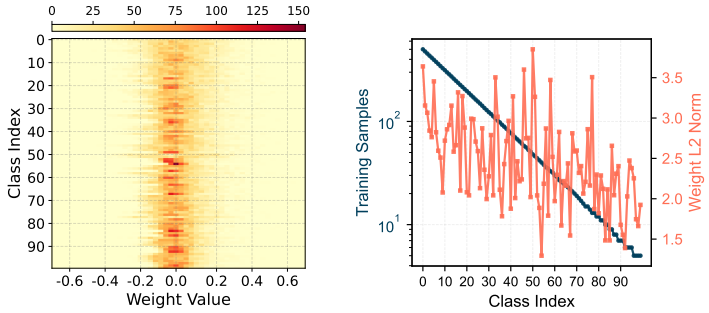
B Additional Analysis of RNR

Figure 9 demonstrates that this strong correlation between weight norm and label distributions also holds for CIFAR-100, a more complex dataset, while Figure 10 shows that RNR similarly disrupts this relationship.



(a) Weight distribution (b) Inferred Distribution

Figure 9: Output layer weight and sample distributions for ResNet-18 trained on long-tailed CIFAR-100 **without RNR**.



(a) Weight distribution (b) Inferred Distribution

Figure 10: Output layer weight and sample distributions for ResNet-18 trained on long-tailed CIFAR-100 **with RNR**.

Figure 11 depicts the RNR training procedure, where random target norms sampled from $\mathcal{N}(\mu, \sigma^2)$ regularize classifier weight magnitudes during local updates.

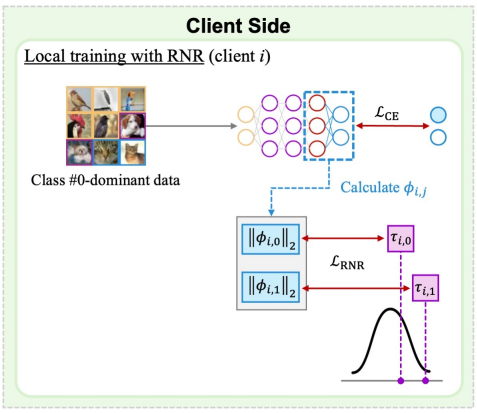


Figure 11: Overview of local training with RNR.

C Implementation Details

To ensure the transparency and reproducibility of our results, we release our code at `https://github.com/lgj413-koreacu/WLIA-RNR`. The code package comprises the preprocessing scripts for the datasets presented in Table 6, data partitioning strategies for simulating non-IID distributions, and the corresponding model configurations with hyperparameter specifications. It also encompasses the implementations of all baseline methods evaluated in this study. Specifically, PGDA and SLIA were implemented using publicly available codebases, while Decaf was carefully reproduced according to the algorithmic details and methodology described in the original paper.

All experiments were performed on a system equipped with two Intel Xeon Gold 6240R CPUs (96 cores in total), 256GB of DDR4 system memory, and two NVIDIA RTX A6000 GPUs (each with 48GB GDDR6 memory). The operating system was Ubuntu 22.04 LTS with CUDA for GPU acceleration. This consistent hardware configuration, combined with fixed random seeds, enabled reliable and fair performance evaluation across all methods.

Table 6: Dataset statistics. The Features column represents the input dimension, corresponding to the number of attributes for tabular data, vocabulary size for text data, and pixel count for image data.

Name	Modality	Samples	Features	Classes
Drive Diagnosis	Tabular	58,509	48	11
DBpedia	Text	630,000	30,522	14
MNIST	Image	70,000	784	10
Fashion-MNIST	Image	70,000	784	10
CIFAR-10	Image	60,000	3,072	10
CIFAR-100	Image	60,000	3,072	100

D Additional Results for Attack

D.1 Overall Performance with Standard Deviation

Tables 7 and 8 report the label distribution inference performance, where all results are averaged over three independent experimental runs with standard deviations provided.

D.2 Attack Performance for MobileViT

Table 9 presents the attack performance on MobileViT using CIFAR-10/100. Under the FedAvg protocol, WLIA consistently outperforms baseline methods. For HtFL methods, while WLIA's performance varies, it generally achieves comparable results to the FedAvg baselines.

D.3 Data Heterogeneity and Client Scalability

Tables 10 and 11 demonstrate that WLIA generally outperforms baseline methods across different settings of data heterogeneity (α) and client population (M). Notably, SLIA exhibits superior performance on several datasets under highly heterogeneous conditions ($\alpha = 0.1$), though no systematic trend emerges across all settings.

Table 7: Label distribution leakage on Drive, DBpedia, and MNIST datasets (mean $\pm$ std over 3 seeds).

Attack	Drive		DBpedia		MNIST	
	TV	KL	TV	KL	TV	KL
Random	0.455 ± 0.001	0.638 ± 0.003	0.453 ± 0.001	0.615 ± 0.002	0.451 ± 0.002	0.600 ± 0.040
PGDA	0.304 ± 0.004	0.355 ± 0.017	0.278 ± 0.006	0.277 ± 0.023	0.589 ± 0.006	1.442 ± 0.015
SLIA	0.222 ± 0.012	0.254 ± 0.013	0.269 ± 0.007	0.442 ± 0.025	0.315 ± 0.001	1.275 ± 0.027
Decaf	0.312 ± 0.019	0.612 ± 0.012	0.423 ± 0.015	0.535 ± 0.019	0.405 ± 0.001	0.538 ± 0.022
WLIA	$\mathbf{0.203 \pm 0.006}$	$\mathbf{0.189 \pm 0.006}$	$\mathbf{0.242 \pm 0.008}$	$\mathbf{0.237 \pm 0.013}$	$\mathbf{0.202 \pm 0.001}$	$\mathbf{0.149 \pm 0.025}$

Table 8: Label distribution leakage on FMNIST, CIFAR-10, and CIFAR-100 datasets (mean $\pm$ std over 3 seeds).

Attack	FMNIST		CIFAR-10		CIFAR-100	
	TV	KL	TV	KL	TV	KL
Random	0.528 ± 0.033	0.817 ± 0.017	0.452 ± 0.002	0.614 ± 0.005	0.482 ± 0.001	0.709 ± 0.001
PGDA	0.575 ± 0.033	1.329 ± 0.096	0.381 ± 0.034	0.603 ± 0.073	0.444 ± 0.021	0.709 ± 0.227
SLIA	0.284 ± 0.008	0.517 ± 0.040	0.215 ± 0.016	0.188 ± 0.023	0.197 ± 0.001	0.127 ± 0.010
Decaf	0.368 ± 0.001	0.617 ± 0.120	0.182 ± 0.016	1.779 ± 0.082	0.391 ± 0.007	0.587 ± 0.018
WLIA	$\mathbf{0.279 \pm 0.011}$	$\mathbf{0.252 \pm 0.020}$	$\mathbf{0.052 \pm 0.001}$	$\mathbf{0.028 \pm 0.003}$	$\mathbf{0.129 \pm 0.001}$	$\mathbf{0.061 \pm 0.002}$

Table 9: Label distribution leakage quantified by total variation distance ($\overline{\text{LDL}}^{\text{TV}}$) and KL divergence ($\overline{\text{LDL}}^{\text{KL}}$) for four federated learning methods on CIFAR-10 and CIFAR-100 with MobileViT.

Attack	FL Method	CIFAR-10		CIFAR-100	
		TV	KL	TV	KL
Random	FedAvg	0.452	0.614	0.482	0.709
PGDA	FedAvg	0.447	1.237	0.557	1.837
SLIA	FedAvg	0.243	0.239	0.127	0.065
Decaf	FedAvg	0.205	2.067	0.468	0.870
WLIA	FedAvg	**0.089**	**0.237**	**0.089**	**0.005**
WLIA	LG-FedAvg	0.125	0.104	0.111	0.051
WLIA	FedClassAvg	0.282	0.765	0.326	0.352
WLIA	CHFL	0.148	0.106	0.315	0.289

Table 10: Comparison of $\overline{\text{LDL}}^{\text{TV}}$ on five datasets with varying heterogeneity (α).

Attack	Drive		DBpedia		MNIST		FMNIST		CIFAR-10	
	0.1	1.0	0.1	1.0	0.1	1.0	0.1	1.0	0.1	1.0
Random	0.729	0.375	0.683	0.404	0.723	0.341	0.682	0.399	0.695	0.381
PGDA	0.289	0.288	0.195	0.264	0.611	0.537	0.435	0.514	0.317	0.509
SLIA	0.274	0.231	**0.184**	0.280	**0.171**	0.355	**0.277**	0.330	0.177	0.287
Decaf	0.304	0.291	0.549	0.371	0.437	0.360	0.321	0.365	0.125	0.215
WLIA	**0.223**	**0.196**	0.317	**0.242**	0.268	**0.171**	0.285	**0.234**	**0.073**	**0.070**

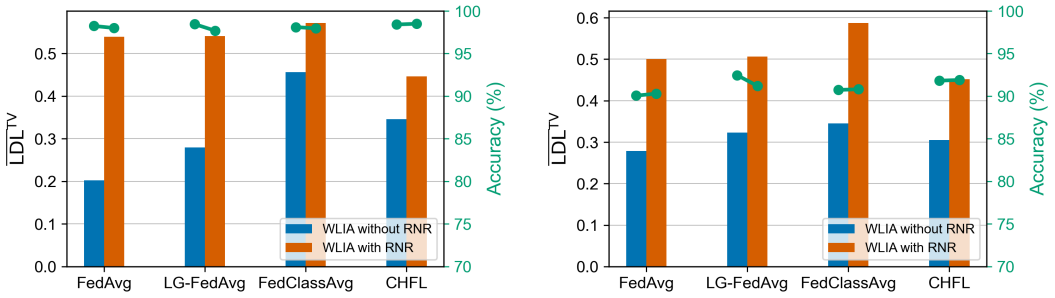
Table 11: Comparison of $\overline{\text{LDL}}^{\text{TV}}$ on five datasets with varying number of clients (M).

Attack	Drive		DBpedia		MNIST		FMNIST		CIFAR-10	
	50	100	50	100	50	100	50	100	50	100
Random	0.531	0.513	0.480	0.516	0.507	0.509	0.517	0.509	0.499	0.499
PGDA	0.247	0.246	0.305	0.326	0.472	0.410	0.364	0.352	0.337	0.350
SLIA	0.273	0.326	0.411	0.408	0.464	0.458	0.418	0.491	0.361	0.407
Decaf	0.284	0.274	0.418	0.471	0.411	0.391	0.344	0.342	0.125	0.129
WLIA	**0.243**	**0.202**	**0.271**	**0.280**	**0.244**	**0.265**	**0.275**	**0.260**	**0.074**	**0.076**

E Additional Results for Defense

E.1 Overall Performance

Figure 12 shows that RNR consistently degrades attack performance across all FL methods on MNIST and FMNIST.

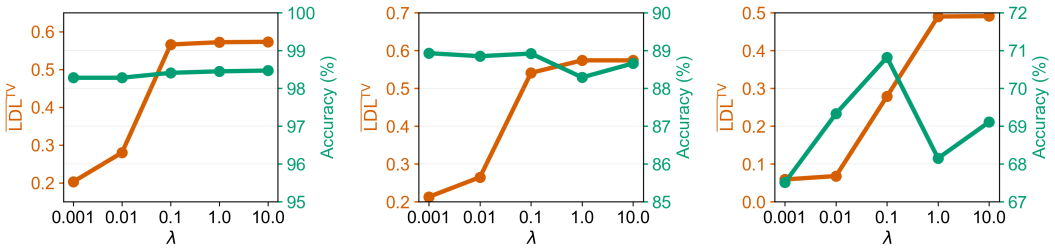


(a) MNIST (b) FMNIST

Figure 12: Defense performance of RNR against WLIA.

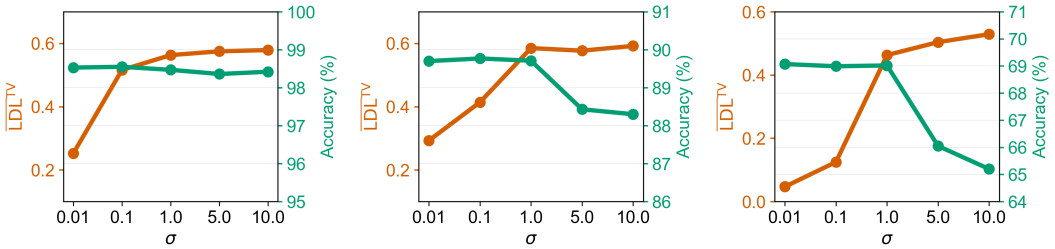
E.2 Hyperparameter Sensitivity Analysis

Figures 13 and 14 illustrate the impact of hyperparameters λ and σ on the defense performance and accuracy under FedAvg with WLIA and RNR.



(a) MNIST (b) FMNIST (c) CIFAR-10

Figure 13: Impact of regularization parameter λ.

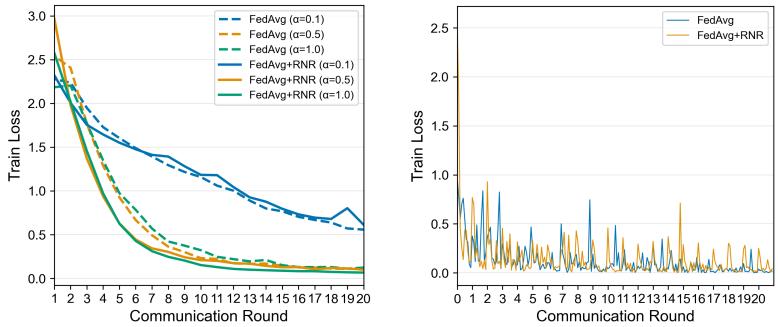


(a) MNIST (b) FMNIST (c) CIFAR-10

Figure 14: Impact of target norm parameter σ.

E.3 Impact of RNR on Convergence

We present convergence curves on the Drive Diagnosis dataset in Figure 15. As shown in Figure 15(a), FedAvg and FedAvg+RNR exhibit similar convergence rates across all heterogeneity levels, confirming that RNR does not hurt global convergence. A similar trend is observed at the client level under high heterogeneity ($\alpha = 0.1$) in Figure 15(b), confirming that RNR does not destabilize local training. The periodic spike pattern is a natural consequence of the FL protocol: at each communication round, the client downloads the updated global model, causing a temporary increase in local loss before training resumes.



(a) Global convergence (b) Client #2 convergence ($\alpha = 0.1$)

Figure 15: Convergence curves of FedAvg and FedAvg+RNR on the Drive Diagnosis dataset.

E.4 Comparison with Differential Privacy

To benchmark RNR against a well-established privacy mechanism, we evaluate (ε, δ)-differential privacy with WLIA, measuring both LDL^{TV} and classification accuracy using the Opacus library [1]. We configure privacy budgets $\varepsilon \in \{99.85, 0.771, 0.130, 0.035\}$ by setting noise multipliers $\sigma \in \{0.25, 1.0, 4.0, 16.0\}$ with failure probability $\delta = 10^{-5}$ under composition. Table 12 presents the comparative analysis across three datasets. While DP was not specifically designed to defend against label distribution leakage, the results demonstrate its fundamental privacy-utility tradeoff: achieving meaningful privacy guarantees against WLIA requires substantial utility degradation. In contrast, RNR, designed specifically for this threat model, effectively mitigates the attack while preserving global model utility.

Table 12: Comparison of WLIA performance under (ε, δ)-DP and RNR for $\overline{\text{LDL}}^{\text{TV}}$ and model accuracy (Acc., %).

Defense	**Drive**		**DBpedia**		**FMNIST**	
	$\overline{\text{LDL}}^{\text{TV}}$	Acc.	$\overline{\text{LDL}}^{\text{TV}}$	Acc.	$\overline{\text{LDL}}^{\text{TV}}$	Acc.
$\varepsilon = 99.85$	0.116	78.92	0.472	59.74	0.303	74.45
$\varepsilon = 0.771$	0.139	77.37	0.483	16.76	0.419	73.50
$\varepsilon = 0.130$	0.327	57.74	0.482	9.65	0.462	62.26
$\varepsilon = 0.035$	0.414	24.88	0.478	8.61	0.467	39.02
RNR	0.458	91.26	0.397	94.98	0.500	90.30

E.5 Applying RNR to Other Attack Methods

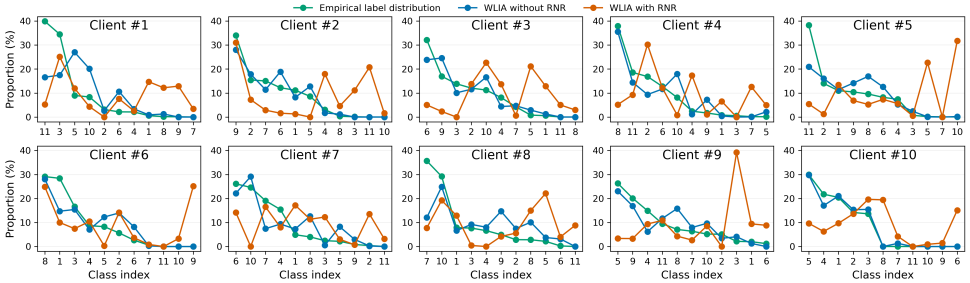
To assess RNR's generalizability, we applied it to three baseline attacks: PGDA, SLIA, and Decaf. Table 13 compares $\overline{\text{LDL}}^{\text{TV}}$ with and without RNR. While RNR generally increases $\overline{\text{LDL}}^{\text{TV}}$ (indicating better defense), the improvements are substantially smaller than those observed against WLIA. Moreover, the defense effectiveness exhibits inconsistent patterns across different settings, with some cases even showing marginal performance degradation. These findings suggest that RNR's defensive capability is specifically effective against weight norm-based attacks like WLIA, but offers limited protection against baseline methods that rely on different inference mechanisms.

Table 13: Comparison of $\overline{\text{LDL}}^{\text{TV}}$ for baseline methods with RNR.

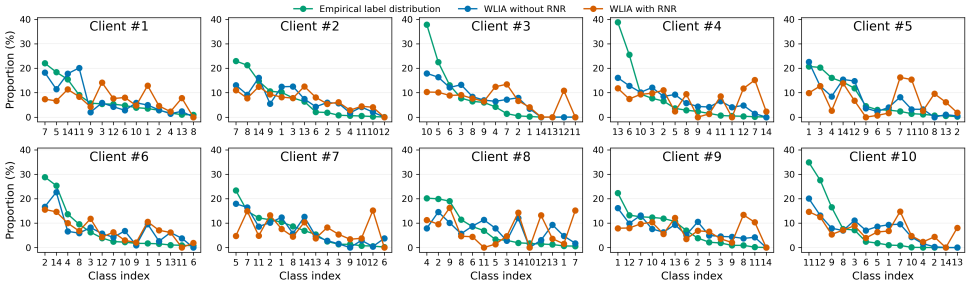
Methods	**Drive**	**DBpedia**	**FMNIST**	**CIFAR-10**
PGDA	0.304	0.278	0.474	0.381
PGDA + RNR	**0.335**	0.268	**0.535**	**0.445**
SLIA	0.222	0.269	0.284	0.215
SLIA + RNR	**0.233**	0.251	**0.303**	0.205
Decaf	0.312	0.423	0.368	0.182
Decaf + RNR	**0.427**	**0.454**	**0.495**	**0.183**

E.6 Per-Client Performance

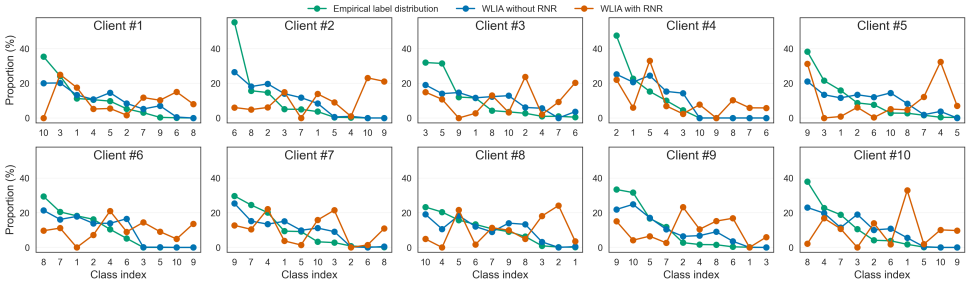
Figures 16 and 17 present per-client label distribution comparisons across five datasets under FedAvg, with classes ranked by decreasing data proportion. Without defense (blue curves), WLIA accurately recovers the ground-truth distribution (green), demonstrating practical inference even on CIFAR-100 despite its 100 classes. However, when RNR is deployed (red curves), the inferred distributions deviate substantially from both the ground truth and the undefended baseline, confirming that RNR successfully disrupts WLIA's inference mechanism.



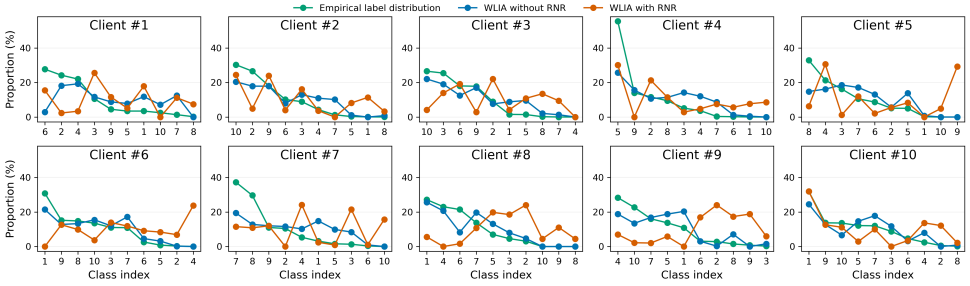
(a) Drive



(b) DBpedia



(c) MNIST



(d) FMNIST

Figure 16: Client-level defense performance of RNR against WLIA under FedAvg on Drive, DBpedia, MNIST, and FMNIST.

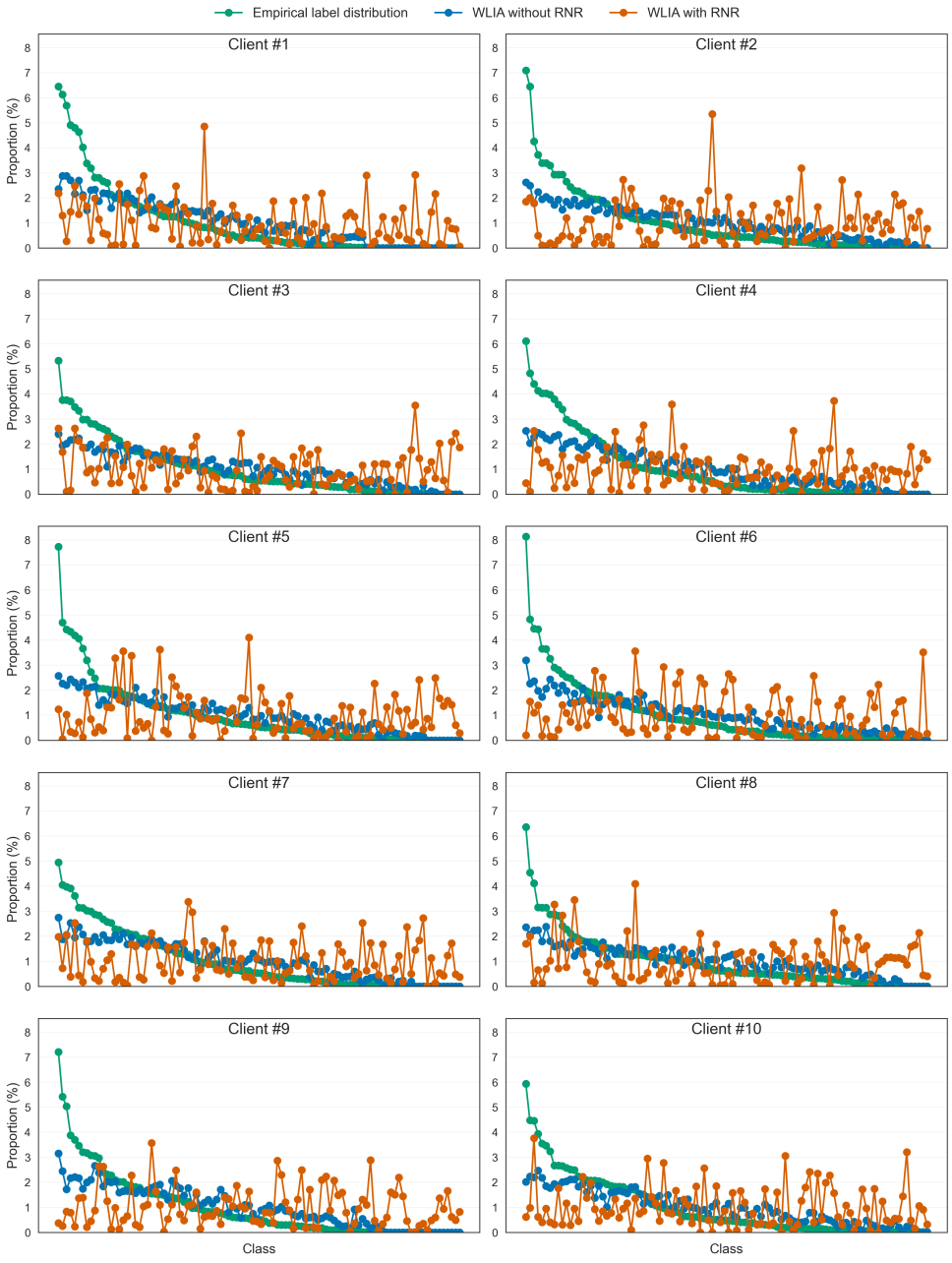


Figure 17: Client-level defense performance of RNR against WLIA under FedAvg on CIFAR-100. The x-axis represents classes ranked by decreasing training sample proportion.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: Described in the Introduction section.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Described in the Limitation and Conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

   Justification: This paper does not include theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: Described in the Experiments section and the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: Described in supplemental material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Described in the Experiments section and supplemental material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: Confidence intervals are reported in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Described in the Experiments section and the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: This paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: The proposed attack method can be applied to preference leakage cyberattacks.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: This paper uses open-source datasets, and we cite the original papers.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: Described in the supplemental material.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: LLMs were used only to correct the grammar of the manuscript.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.