Confusion-Resistant Federated Learning via Diffusion-Based Data Harmonization on Non-IID Data

Xiaohong Chen[1,2,3] **Canran Xiao**[1*] **Yongmei Liu**[1,4]
[1] School of Business, Central South University, Changsha, Hunan 410083, China
[2] Xiangjiang Laboratory, Changsha, Hunan 410205, China
[3] School of Advanced Interdisciplinary Studies, School of Management Science and Engineering, Hunan University of Technology and Business, Changsha, Hunan 410205, China
[4] Urban Smart Governance Laboratory, Changsha, Hunan 410083, China
c88877803@163.com, xiaocanran@csu.edu.cn, liuyongmeicn@163.com

Abstract

Federated learning has become a pivotal distributed learning paradigm, involving collaborative model updates across multiple nodes with private data. However, handling non-i.i.d. (not identically and independently distributed) data and ensuring model consistency across heterogeneous environments present significant challenges. These challenges often lead to model performance degradation and increased difficulty in achieving effective communication among participant models. In this work, we propose Confusion-Resistant Federated Learning via Consistent Diffusion (CRFed), a novel framework designed to address these issues. Our approach introduces a new diffusion-based data harmonization mechanism that includes data augmentation, noise injection, and iterative denoising to ensure consistent model updates across non-i.i.d. data distributions. This mechanism aims to reduce data distribution disparities among participating nodes, enhancing the coordination and consistency of model updates. Moreover, we design a confusion-resistant strategy leveraging an indicator function and adaptive learning rate adjustment to mitigate the adverse effects of data heterogeneity and model inconsistency. Specifically, we calculate importance sampling weights based on the optimal sampling probability, which guides the selection of clients and the sampling of their data, ensuring that model updates are robust and aligned across different nodes. Extensive experiments on benchmark datasets, including MNIST, FashionMNIST, CIFAR-10, CIFAR-100, and NIPD, demonstrate the effectiveness of CRFed in improving accuracy, convergence speed, and overall robustness in federated learning scenarios with severe data heterogeneity.

1 Introduction

Federated Learning (FL) [McMahan et al., 2017b] has emerged as a powerful paradigm for distributed machine learning, enabling multiple clients to collaboratively train a shared model without exchanging raw data. This approach addresses critical concerns around data privacy and security, which are increasingly significant in various sectors such as healthcare [Antunes et al., 2022], finance [Chatterjee et al., 2023], and IoT [Li et al., 2020a, Pan et al., 2023, Yao et al., 2024]. However, one of the fundamental challenges in FL is dealing with non-independent and identically distributed (non-IID) data, which can significantly impair the performance and convergence of the global model [Zhu et al., 2021].

*Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

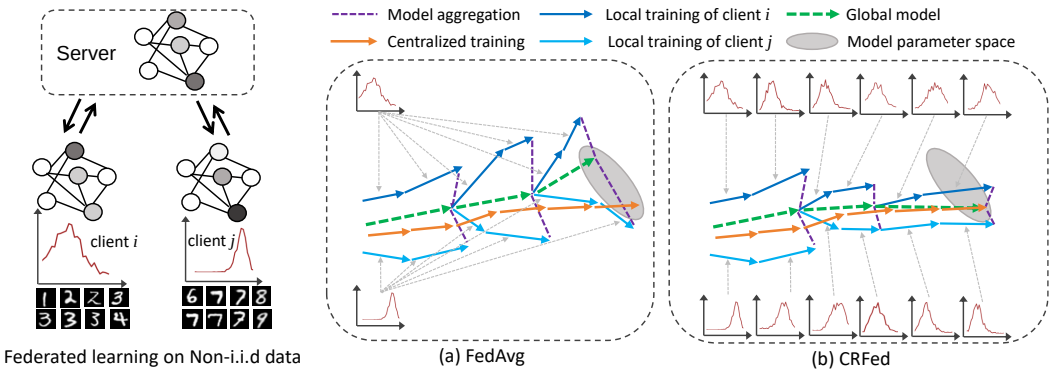

Figure 1: Problem illustration of federated learning on Non-i.i.d data.

As illustrated in Figure 1, FL on non-IID data often suffers from issues like divergent model updates and inconsistent global models. Client models trained on heterogeneous data distributions tend to diverge [Ye et al., 2023], making it difficult for the server to aggregate them into a coherent global model. This divergence is due to inconsistencies in data sources [Xiao and Liu, 2024] and distributions across clients [Duan et al., 2021]. This problem leads to reduced accuracy and slower convergence rates, highlighting the need for effective solutions to handle data heterogeneity. Existing research has made significant strides in improving the robustness and efficiency of FL in non-IID settings. Notable methods include FedProx [Li et al., 2020b], which adds a proximal term to handle heterogeneity. Techniques such as MOON [Li et al., 2021b]and FedGen [Nguyen et al., 2021] introduce sophisticated strategies like contrastive learning and data generation to mitigate the effects of data heterogeneity. Despite these advancements, issues related to data distribution disparities and model inconsistency persist, limiting the scalability and effectiveness of FL in real-world scenarios.

Given the critical gaps in existing FL approaches, particularly their limited robustness to severe non-IID data distributions, there is a pressing need for more resilient and adaptive solutions. This research is motivated by the necessity to enhance FL's capability to handle heterogeneous data efficiently. The primary objective of this study is to develop a novel FL framework, Confusion-Resistant Federated Learning via Consistent Diffusion (CRFed), which integrates advanced mechanisms to address data distribution disparities and enhance model consistency across clients.

Our contributions can be summarized as follows:

1. We propose a novel indicator function that dynamically adjusts sample weighting based on loss values and uncertainties, facilitating a self-paced learning approach that prioritizes more difficult samples over time.
2. Our framework employs a diffusion-based mechanism to harmonize data distributions, involving iterative noise injection and denoising processes that align local data with the desired distribution.
3. We implement a strategic client selection method based on the indicator function, ensuring the inclusion of the most reliable clients, which enhances the robustness and consistency of model updates.
4. Our extensive experimental evaluations demonstrate that CRFed achieves state-of-the-art performance on benchmark datasets. CRFed outperforms existing methods significantly in terms of accuracy and convergence speed under various non-IID settings.

2 Related Work

2.1 Non-IID Challenge in Federated Learning

The issue of non-IID data in FL was initially highlighted by FedAVG [McMahan et al., 2017a], and it has since been demonstrated that this challenge can significantly hinder the convergence and overall performance of the global model [Zhao et al., 2018, Li et al., 2019]. Numerous studies, categorized as client-centric methods, have been proposed to tackle this problem by adjusting the local training

objectives using insights from the global model and the local models of other clients [Wang et al., 2021]. For instance, FedProx [Li et al., 2020b] introduced a proximal term to constrain local updates by leveraging the global model. SCAFFOLD [Karimireddy et al., 2020] utilized control variates to correct for local training drift, while FedDyn [Acar et al., 2021] introduced a dynamic regularizer for parallelizing gradients among clients. MOON [Li et al., 2021b] applied contrastive learning to minimize the discrepancy between model representations, thereby correcting local training.

Despite their contributions, these methods fall short of fully resolving the core of the non-IID issue and may experience performance limitations in scenarios with highly skewed data distributions [Li et al., 2022]. Beyond client-side adjustments, the server also plays a role in mitigating the adverse effects of non-IID data by calibrating the biased global model post-aggregation. For example, CCVR [Luo et al., 2021] uses virtual representations from an approximated Gaussian mixture model to correct the classifier. FedFTG [Zhang et al., 2022] employs data-free knowledge distillation to refine the global model with the knowledge derived from local models. Additionally, strategies such as client clustering [Ghosh et al., 2020, Long et al., 2023] and client selection [Zhang et al., 2021, Wang et al., 2020], can be implemented by the server to alleviate the non-IID problem. IFCA [Ghosh et al., 2020] iteratively estimates client cluster identities based on local empirical loss and updates model parameters for each cluster via gradient descent.

2.2 Importance Sampling in Federated Learning

In federated learning (FL), data sampling strategies are vital for enhancing distributed training efficiency. [Tuor et al., 2020] proposed selecting local training data based on user-end data correlation analysis. This led to dynamic sampling strategies like [Li et al., 2021a], where training sample importance is determined by model gradient magnitudes. Similarly, [Rizk et al., 2022] used gradient norms to derive sampling weights, minimizing theoretical convergence bounds. However, these methods require immediate gradient computations, increasing local overhead, and assume convex loss functions, which may not apply to deep learning models [Rizk et al., 2021]. Therefore, developing importance sampling methods suitable for deep learning-based FL remains an open challenge.

FL convergence can be theoretically analyzed due to the model aggregation mechanism [98, 101-102, 150], with experimental validation for deep learning tasks [Wan et al., 2021]. Most studies rely on theoretical derivations, limiting practical application. This study aims to use a diffusion model to automate the modeling of optimal sampling strategies in FL.

3 Method

3.1 Overview

The CRFed framework, shown in Figure 2, addresses challenges posed by non-i.i.d. data in FL. Our approach integrates a diffusion mechanism and a confusion-resistant strategy to ensure consistent and robust model updates across heterogeneous data distributions. The core idea of CRFed is that the performance of client i's data on the global model reflects its contribution to the training process. By using an optimal indicator function, we determine the optimal data sampling probability for each client, enhancing training efficiency and model performance.

The framework comprises several key components: the current global model downloaded by clients at time t; the Model Encoder and Meta-model, which process the global model and client-specific data for the diffusion process; an indicator function, computed using client i's data on the global model; the Diffusion-based Data Harmonization Mechanism, which uses data augmentation, noise injection, and probabilistic modeling to mitigate data distribution disparities; and the Distribution Decoder, which aligns the denoised data distribution with the desired distribution.

3.2 Indicator Function and Meta-model

The Indicator Function $I_\lambda(l_i, \sigma_i)$ is designed to measure the reliability of the i-th sample's loss value l_i and its associated uncertainty σ_i. The design is inspired by self-paced learning [Fan et al., 2017, Castells et al., 2020], which adjusts the weights of samples based on their loss values, allowing for a gradual learning process from easy to difficult samples. In the context of federated learning, this means that each client should adopt a self-paced learning paradigm, sampling its data in a way that allows the global model to learn from simple to complex tasks. The Indicator Function captures this performance and is defined as follows:

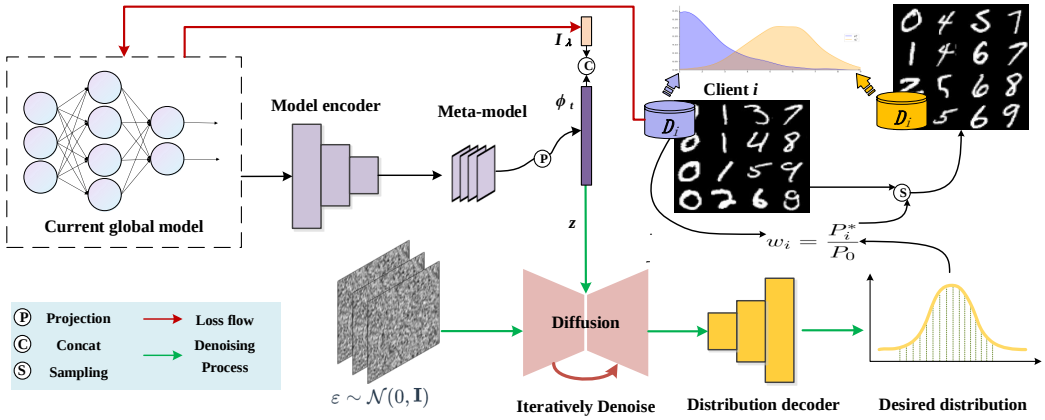

Figure 2: **CRFed Framework.** The process begins with the current global model, which is downloaded by clients. The model encoder processes the global model, and the meta-model is obtained. This meta-model is then projected into a higher-dimensional space and concatenated with the indicator function, forming the combined representation z_i. The diffusion-based data harmonization mechanism adds noise to this representation and iteratively denoises it to achieve the desired distribution. The distribution decoder then aligns the denoised data distribution. Client i's data is sampled based on importance sampling weights w_i, calculated as the ratio of the optimal sampling probability P_i^* to the original data distribution P_0. This ensures that the sampled data aligns with the desired distribution, following a curriculum learning approach that progresses from easy to difficult samples, thus enhancing overall model performance.

$$I_\lambda(l_i, \sigma_i) = (l_i - \tau)\sigma_i + \lambda(\log \sigma_i)^2 \tag{1}$$

where λ is a pre-set regularization coefficient. τ is a confidence threshold that determines the difficulty of the sample based on its loss value. It can either be a fixed constant or a dynamically adjusted weighted average during the training process.

The Indicator Function can be further explained using the following steps: for each client i, the loss value l_i of each sample is calculated on the current global model, and the uncertainty σ_i of each sample is estimated based on its difficulty. The Indicator Function $I_\lambda(l_i, \sigma_i)$ is then used to assign weights to the samples, with easier samples having lower weights and more difficult samples having higher weights. This adaptive weighting mechanism ensures that the model focuses more on difficult samples over time, leading to improved learning efficiency and robustness.

Theorem 3.1. *In the CRFed framework, using the indicator function $I_\lambda(l_i, \sigma_i)$ ensures stable and convergent updates for heterogeneous federated learning. For an appropriately chosen learning rate η, the model update rule for client i at iteration t,*

$$\theta_{t+1} = \theta_t - \eta \left(\sigma_i^* + (l_i - \tau)k + 2\lambda \frac{\log \sigma_i^*}{\sigma_i^*} k \right) \nabla_\theta l_i, \tag{2}$$

guarantees a decreasing step size, promoting convergence. Moreover, CRFed achieves a tighter bound on update steps than FedAvg, indicating faster convergence under the same conditions.

The proof of Theorem 3.1 can be found in Appendix A.2.

Given the global model θ, the optimal uncertainty σ_i^* can be derived through the following theorem:

Theorem 3.2. *The optimal uncertainty σ_i^* for a given loss value l_i is obtained by minimizing the Indicator Function $I_\lambda(l_i, \sigma_i)$. The solution is given by:*

$$\sigma_i^*(l_i) = \exp\left(-W\left(\frac{1}{2\lambda} \max\left(-\frac{2}{e}, l_i - \tau \right) \right) \right) \tag{3}$$

where $W(\cdot)$ is the Lambert W function.

We rewrite the indicator function by setting $\sigma_i = e^{x_i}$, reduce the derivative condition to a Lambert W form under domain constraints, and thus obtain a closed-form solution for σ_i^*; the complete derivation is presented in Appendix A.3.

For each sample indexed by i, we determine the optimal uncertainty σ_i^* as a function of its loss l_i. Specifically, let $I_\lambda(l_i, \sigma_i)$ be the indicator function defined before. Then, selecting σ_i^* that minimizes $I_\lambda(l_i, \sigma_i)$ induces an adaptive weighting scheme:

$$\omega_i(l_i) \;\propto\; \frac{1}{\sigma_i^*(l_i)}, \tag{4}$$

ensuring that when l_i is relatively large, the corresponding optimal σ_i^* decreases, thus increasing $\omega_i(l_i)$ and emphasizing more difficult samples. Conversely, smaller l_i values lead to larger σ_i^* and lower $\omega_i(l_i)$, indicating that easier samples receive diminished focus. As a result, the distribution defined by σ_i^* is optimally aligned with the current global model θ_t, conforming to the self-paced learning principle.

Since $I_\lambda(l_i, \sigma_i)$ can be regarded as quantifying each client's contribution relative to θ_t, we embed θ_t within the diffusion model as follows: using an autoregressive encoder E, we compress θ_t to a meta-model $\phi_t = E(\theta_t)$. Subsequently, ϕ_t is projected into a high-dimensional representation $P(\phi_t)$, and concatenated with $I_\lambda(l_i, \sigma_i)$ to form

$$z_i \;=\; \mathrm{concat}\big(P(\phi_t),\, I_\lambda(l_i, \sigma_i)\big). \tag{5}$$

We then feed z_i into the diffusion model, denoted by $\mathrm{DiffusionModel}(z_i)$, to refine the data distribution with respect to the global context. Further implementation details about the encoder E are provided in Appendix A.4.

3.3 Diffusion-based Data Harmonization

The diffusion-based data harmonization mechanism is a critical component of the CRFed framework, responsible for mitigating data distribution disparities and ensuring consistent model updates across heterogeneous environments. The harmonization process is shown in Figure 3, which involves adding noise to the data distribution and then iteratively denoising it to achieve the desired distribution. The workflow of this mechanism can be divided into two main processes: the forward diffusion process and the reverse denoising process.

Forward Diffusion Process Suppose that a training sample $\mathbf{x}_0$ is of a certain distribution, denoted as $q(\mathbf{x}_0)$. In the forward diffusion process, Gaussian noise with variance $\beta_t \in (0, 1)$ is added gradually to the sample $\mathbf{x}_0$ for T steps, resulting in a latent sample $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$. The process is defined as follows:

$$q(\mathbf{z}_{1:T}|z_i) = \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{z}_{t-1}), \tag{6}$$

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t\mathbf{I}). \tag{7}$$

Using notations $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, the sample $\mathbf{z}_t$ can be defined directly as:

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t} z_i + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}). \tag{8}$$

Reverse Denoising Process The reverse denoising process aims to sample reversely from $\mathbf{z}_T$ through transition probabilities $q(\mathbf{z}_{t-1}|\mathbf{z}_t)$ for timesteps $T-1$ through 1, yielding a sample drawn from $q(z_i)$. The transition $q(\mathbf{z}_{t-1}|\mathbf{z}_t)$ is a Gaussian distribution, tractable when conditioned on z_i:

$$q(\mathbf{z}_{t-1}|\mathbf{z}_t, z_i) = \mathcal{N}(\mathbf{z}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{z}_t, z_i), \tilde{\beta}_t\mathbf{I}), \tag{9}$$

where the mean $\tilde{\boldsymbol{\mu}}_t$ and variance $\tilde{\beta}_t$ are calculated as:

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{z}_t, z_i) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_t \right), \tag{10}$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t. \tag{11}$$

The reverse transition probability $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t)$ relies on the entire data distribution and is approximated through a neural network:

$$p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{z}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{z}_t, t)), \tag{12}$$

The detailed derivations and computations for these processes can refer to A.1. In the context of federated learning, the reverse denoising process starts from the optimal indicator function I^* obtained in the forward diffusion process. By progressively denoising, we obtain the optimal sampling probability for client i, ensuring that the final data distribution aligns with the desired distribution. This minimizes the impact of data heterogeneity and ensures robust model updates across all clients.

3.4 Confusion-Resistant Strategy

The confusion-resistant strategy is designed to address the challenges posed by data heterogeneity and model inconsistency in federated learning. It consists of three key components: client selection based on the indicator function, data sampling using the diffusion-based harmonization mechanism, and adaptive learning rate adjustment.

Client Selection Strategy To mitigate the adverse effects of data heterogeneity, we select clients based on their indicator function $I_\lambda(l_i, \sigma_i)$, which quantifies the reliability of their data. Clients with the lowest indicator values, reflecting higher data reliability, are chosen for training. This approach follows the curriculum learning paradigm, where lower values indicate better data:

$$\text{Selected Clients} = \{i | I_\lambda(l_i, \sigma_i) \leq \gamma\}, \tag{13}$$

where γ is a dynamically adjusted threshold ensuring the selection of the most suitable clients.

Data Sampling Strategy For each selected client, the optimal sampling probability P_i^* is determined through the reverse denoising process, starting from the optimal indicator function I^*. This ensures that the sampled data aligns with the desired distribution, enhancing the robustness of model updates. The importance sampling weight w_i is calculated as follows:

$$w_i = \frac{P_i^*}{P_0}. \tag{14}$$

where P_0 is the original data distribution. Using w_i, we sample the local training data ($\mathcal{D}_i^{\text{sampled}} = \text{Sample}(\mathcal{D}_i, w_i)$). This sampling ensures the effective sampling probability aligns with P_i^*.

The distribution decoder, which is implemented as an autoencoder, is then used to decode the denoised data distribution. The autoencoder is trained to map the denoised samples back to the desired distribution, further ensuring that the data used for training is aligned with the ideal distribution. Refer to the A.4 for more details of distribution decoder.

Adaptive Learning Rate Adjustment The learning rate η_i for each client is adjusted based on the indicator function value, enhancing the influence of more reliable data:

$$\eta_i = \eta_0 \cdot \frac{I_\lambda(l_i, \sigma_i)}{\max_j I_\lambda(l_j, \sigma_j)}, \tag{15}$$

where η_0 is the base learning rate.

The complete computational process(pseudocode) of CRFed is provided in the A.5.

4 Experiments

4.1 Experiment Setup

Datasets Our experiments are conducted on four widely used benchmark datasets: MNIST [LeCun et al., 1998], Fashion-MNIST [Xiao et al., 2017], CIFAR-10 [Krizhevsky et al., 2009], and CIFAR-100 [Krizhevsky et al., 2009]. To simulate Non-IID data scenarios, we utilize the Dirichlet distribution [Yurochkin et al., 2019] to generate non-IID partitions with varied concentration parameters, β. Smaller values of β lead to more imbalanced data distributions among clients, thereby increasing levels of data heterogeneity. In our experiments, we set β to 0.5 to reflect this imbalance. In all experiments, we simulate a federated learning environment with 10 edge nodes, i.e., $K = 10$. For the MNIST and FashionMNIST datasets, each node has 600 data samples. For the CIFAR-10 dataset, each node has 500 data samples. For CIFAR-100, the partitioning strategy remains the same, ensuring that each client's local data distribution varies significantly, simulating real-world federated learning scenarios. MNIST and FashionMNIST datasets consist of grayscale images of size 28×28 pixels, with 10 classes. CIFAR-10 and CIFAR-100 contain color images of size 32×32 pixels.

Additionally, we use the NIPD dataset [Yin et al., 2023], a benchmark specifically designed for federated learning in person detection tasks with Non-IID data. This dataset provides a real-world non-IID scenario to test the generalization of CRFed.

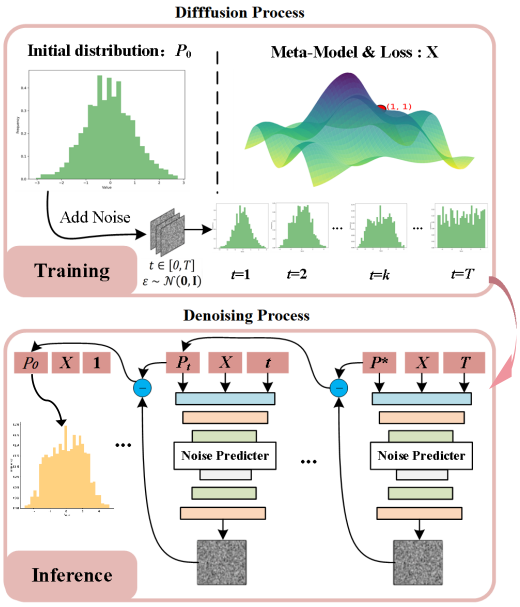

Figure 3: The diffusion-based data harmonization mechanism in CRFed framework. The process involves a forward diffusion process where Gaussian noise is added to the initial data distribution, transforming it into a latent representation. This is followed by a reverse denoising process that iteratively removes the noise, aligning the data distribution with the desired target distribution.

Competing Methods Apart from FedAvg [McMahan et al., 2017a], we compare the proposed algorithm with several benchmarking FL algorithms specialized for solving the non-IID problem, including FedProx [Li et al., 2020b], MOON [Li et al., 2021b], and FedGen [Nguyen et al., 2021]. We also compare our method against HFMDS-FL [Li et al., 2024], FRAug [Chen et al., 2023], G-FML [Yang et al., 2023], FedCD [Long et al., 2023], FedNP [Wu et al., 2023], and FedDPMS [Chen and Vikalo, 2023], which are recent state-of-the-art approaches addressing non-IID data issues in federated learning.

Hyperparameter For local training, the settings are as follows: MNIST with $E = 5$, $B = 10$, $\eta = 5 \times 10^{-3}$; FashionMNIST with $E = 5$, $B = 100$, $\eta = 2 \times 10^{-4}$; CIFAR-10 and CIFAR-100 with $E = 5$, $B = 100$, $\eta = 1 \times 10^{-4}$. Momentum optimization with a coefficient of 0.5 is applied. In the CRFed framework, key hyperparameters include maximum global rounds (T_G) set to 100, local training cycles (E_l) per global round set to 1, regularization coefficient (λ) set to 0.1, dynamically adjusted confidence threshold (τ), and client selection threshold (γ) initially set to 0.5. These parameters are fine-tuned based on preliminary experiments to ensure training efficiency and model performance. The experiments were conducted using an NVIDIA GeForce RTX 4060 GPU, which has 8GB of VRAM. Detailed configurations of model structure are provided in A.6.

Evaluation metrics The primary evaluation metrics for our experiments focus on accuracy and the number of training rounds needed to reach convergence, addressing the challenges posed by non-IID data in federated learning. Accuracy is measured at the same training round across different models to ensure fair comparison. Convergence is assessed by the number of rounds required to achieve a target accuracy, which reflects the model's stability and efficiency. For the NIPD dataset,

we use mean Average Precision (mAP) as the evaluation metric. Following [Wang et al., 2020], all reported results are averaged over five runs with different random seeds to account for variability.

4.2 Performance Comparison

Table 1: Test accuracy of CRFed and the competing methods on five datasets. We run five trials with different random seeds and report the mean accuracy.

Scheme	MNIST	FashionMNIST	CIFAR-10	CIFAR-100	NIPD (mAP)
FedAvg [McMahan et al., 2017a]	0.976	0.847	0.650	0.362	0.821
FedProx [Li et al., 2020b]	0.978	0.844	0.655	0.365	0.826
MOON [Li et al., 2021b]	0.980	0.846	0.674	0.372	0.836
FedGen [Nguyen et al., 2021]	0.982	0.862	0.672	0.369	0.841
HFMDS-FL [Li et al., 2024]	0.982	0.868	0.678	0.377	0.846
FRAug [Chen et al., 2023]	0.981	0.865	0.675	0.374	0.851
G-FML [Yang et al., 2023]	0.983	0.870	0.681	0.378	0.854
FedCD [Long et al., 2023]	0.982	0.867	0.677	0.376	0.861
FedNP [Wu et al., 2023]	0.982	0.869	0.680	0.377	0.863
FedDPMS [Chen and Vikalo, 2023]	0.983	0.876	0.680	0.386	0.871
CRFed	**0.985**	**0.878**	**0.683**	**0.389**	**0.882**

Accuracy comparison Table 1 presents the test accuracy of CRFed compared to several federated learning algorithms under a highly heterogeneous setting ($\beta = 0.5$). Our proposed method shows notable improvements over FedAvg [McMahan et al., 2017a], with relative gains of 0.9% on MNIST, 3.7% on FashionMNIST, 5.1% on CIFAR-10, 7.5% on CIFAR-100, and a significant 7.4% improvement in mAP on the NIPD dataset. This highlights CRFed's robustness in handling non-IID data distributions. CRFed consistently outperforms all other methods across the datasets, underscoring its effectiveness in federated learning scenarios with severe data heterogeneity.

Effect of Data Heterogeneity We analyze the impact of data heterogeneity on the performance of the top 5 models by varying the Dirichlet concentration parameter β. Table 2 shows the performance of these models on CIFAR-100 and NIPD datasets for β values ranging from 0.1 to 0.5. As expected, the performance generally decreases with smaller β values due to increased data heterogeneity. Table 2 indicates that as β decreases, representing higher data heterogeneity, the performance of all models declines. CRFed consistently outperforms other methods across different β settings, demonstrating its robustness in handling data heterogeneity. Notably, the relative performance gap between CRFed and other methods widens as β decreases, highlighting its efficacy in more challenging federated learning scenarios.

Table 2: Performance of top 5 models on CIFAR-100 and NIPD datasets under different β values.

	CIFAR-100			NIPD (mAP)		
Scheme	0.1	0.3	0.5	0.1	0.3	0.5
FedDPMS	0.270	0.330	0.386	0.751	0.810	0.871
FRAug	0.268	0.328	0.374	0.746	0.800	0.851
G-FML	0.265	0.329	0.378	0.748	0.802	0.854
FedCD	0.275	0.333	0.376	0.750	0.808	0.861
CRFed	**0.280**	**0.345**	**0.389**	**0.760**	**0.820**	**0.882**

Effect of Increasing Edge Nodes Figure 4 presents the performance of the top 5 models on CIFAR-100 and NIPD datasets as the number of edge nodes K increases from 10 to 100. Across all models, performance generally improves with higher K values, reflecting better data utilization. Notably, CRFed shows the most significant gains, with accuracy increasing from 0.389 to 0.425 on CIFAR-100 and mAP from 0.882 to 0.920 on NIPD. This demonstrates CRFed's superior scalability and effectiveness in handling more edge nodes, making it robust in federated learning environments with increasing data sources.

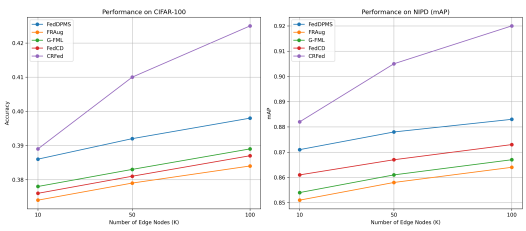

Figure 4: Effect of Increasing Edge Nodes

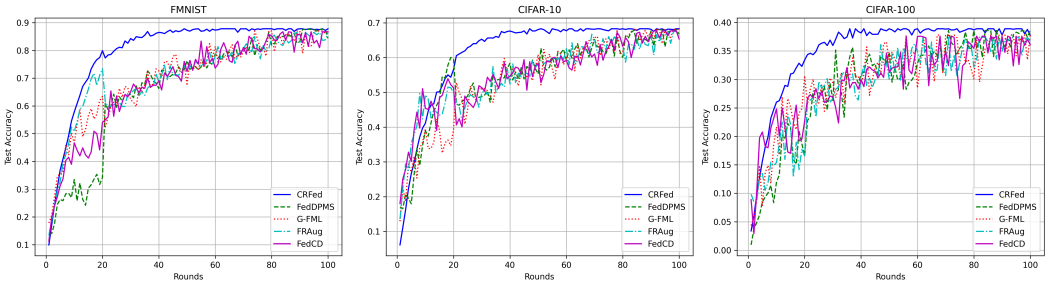

Figure 5: Test accuracy across federated training rounds for top 5 models on FMNIST, CIFAR-10, and CIFAR-100 datasets.

Convergence Rate The convergence performance of the top five models on FMNIST, CIFAR-10, and CIFAR-100 datasets is depicted in Figure 5. As observed, CRFed demonstrates significantly faster and more stable convergence compared to the competing methods across all three datasets. This superior performance is attributed to the diffusion-based data harmonization mechanism, which effectively aligns data distributions, and the confusion-resistant strategy that selects reliable clients and adaptively adjusts learning rates, ensuring efficient and robust training even in highly heterogeneous environments.

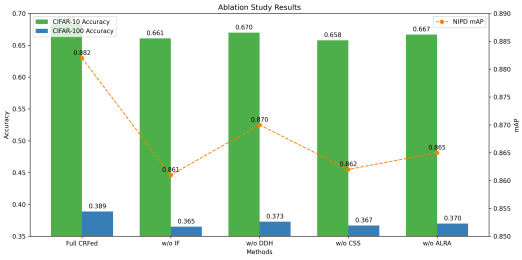

Figure 6: Ablation study results on CIFAR-10, CIFAR-100, and NIPD datasets. The bar charts show the accuracy on CIFAR-10 and CIFAR-100 datasets, while the line plot represents the mAP on the NIPD dataset.

4.3 Ablation Study

To evaluate the contribution of each component in the CRFed framework, we conduct an ablation study by removing or altering specific components and observing the impact on model performance. Removing the Indicator Function and using uniform sampling led to significant performance drops, with CIFAR-10 accuracy falling from 0.683 to 0.661, CIFAR-100 from 0.389 to 0.365, and NIPD mAP from 0.882 to 0.861. Excluding the Diffusion-based Data Harmonization (DDH) mechanism resulted in reduced accuracy on CIFAR-10 (0.670), CIFAR-100 (0.373), and NIPD mAP (0.870), highlighting its role in aligning data distributions. Replacing strategic client selection with random selection markedly decreased performance, emphasizing the importance of reliable client selection. Fixing the learning rate instead of adapting it slowed convergence and destabilized training. These findings validate the theoretical and practical significance of our proposed components in improving federated learning performance.

4.4 Comparison with Importance Sampling Methods

Previous importance sampling methods typically require prior analysis of the data relevance at each client-side [Hsu et al., 2020, Tian et al., 2022] or necessitate deriving optimal sampling weights based on assumptions such as the convexity of the loss function [Rizk et al., 2022, Zhu et al., 2024]. While these methods offer strong theoretical guarantees, they are somewhat limited in their adaptability to real-world federated learning (FL) scenarios. For instance, both FedIR [Hsu et al., 2020] and Harmony [Tian et al., 2022] assume that the server has knowledge of the local distributions of all clients. Although this assumption does not violate the privacy-preserving principles of FL, it can be challenging to obtain in real-world applications.

In contrast, our CRFed does not depend on these assumptions. Instead, it iteratively adjusts the data distributions during the FL process itself, enabling the model to dynamically harmonize the diverse, non-IID data across clients without requiring explicit distributional assumptions or centralized access to all client data distributions. Guided by the indicator function, our CRFed can derive the optimal sampling strategy for each local node.

Moreover, as shown in Table 3, empirical experiments demonstrate that the diffusion model achieves superior performance, outperforming other benchmark methods.

It is worth noting that this comparison is not entirely fair, as each importance sampling method operates under different assumptions. For example, ISFL requires a validation set to update the empirical gradient Lipschitz constants for each local model, while FedIR requires all clients to upload the conditional distribution of images given class labels to match the target distribution. Nevertheless, our CRFed outperforms the others even under less restrictive conditionsunlike ISFedAvg and ISFL, it does not require assumptions about the loss function or gradient variance, and unlike FedIR and Harmony, it does not require centralized access to all client data distributions before calculating the importance sampling weights.

Table 3: The performance of different importance sampling methods on CIFAR-100 under various β values.

	CIFAR-100		
Method	$\beta = 0.1$	$\beta = 0.3$	$\beta = 0.5$
ISFedAvg	0.232	0.285	0.305
ISFL	0.237	0.296	0.314
FedIR	0.258	0.311	0.352
Harmony	0.246	0.313	0.354
CRFed	**0.280**	**0.345**	**0.389**

5 Conclusion

In conclusion, this study tackles the pressing challenge of handling non-i.i.d. data in federated learning environments. We propose the Confusion-Resistant Federated Learning via Consistent Diffusion (CRFed) framework. This framework introduces a novel Indicator Function that dynamically adjusts sample weighting, facilitating a self-paced learning paradigm that prioritizes more difficult samples over time. Additionally, our diffusion-based data harmonization mechanism ensures consistent and aligned data distributions through iterative noise injection and denoising processes, mitigating the adverse effects of data heterogeneity. Our strategic client selection method, guided by the Indicator Function, ensures that the most reliable clients are chosen for training, thus improving the robustness and consistency of global model updates.

Despite the promising results, our approach has certain limitations. The reliance on complex diffusion mechanisms and adaptive strategies may introduce computational overhead, which could be a concern for resource-constrained environments. Future work should focus on optimizing the computational efficiency of the CRFed framework and exploring its applicability to a broader range of real-world federated learning scenarios [Zhang et al., 2023].

6 Acknowledgement

This research was funded by the Basic Science Center Project for National Natural Science Foundation of China (Grant No: 72088101), the Xiangjiang Laboratory Major Project(Grant No: 23XJ01007), and the Fundamental Research Funds for the Central Universities of Central South University(Grant No: 1053320214050).

References

D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

R. S. Antunes, C. André da Costa, A. Küderle, I. A. Yari, and B. Eskofier. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–23, 2022.

T. Castells, P. Weinzaepfel, and J. Revaud. Superloss: A generic loss for robust curriculum learning. *Advances in Neural Information Processing Systems*, 33:4308–4319, 2020.

P. Chatterjee, D. Das, and D. B. Rawat. Federated learning empowered recommendation model for financial consumer services. *IEEE Transactions on Consumer Electronics*, 2023.

H. Chen and H. Vikalo. Federated learning in non-iid settings aided by differentially private synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5026–5035, 2023.

H. Chen, A. Frikha, D. Krompass, J. Gu, and V. Tresp. Fraug: Tackling federated learning with non-iid features via representation augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4849–4859, 2023.

M. Duan, D. Liu, X. Ji, Y. Wu, L. Liang, X. Chen, Y. Tan, and A. Ren. Flexible clustered federated learning for client-level data distribution shift. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):2661–2674, 2021.

Y. Fan, R. He, J. Liang, and B. Hu. Self-paced learning: An implicit regularization perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

A. Ghosh, J. Chung, D. Yin, and K. Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020.

T.-M. H. Hsu, H. Qi, and M. Brown. Federated visual classification with real-world data distribution. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 76–92. Springer, 2020.

S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.

A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

A. Li, L. Zhang, J. Tan, Y. Qin, J. Wang, and X.-Y. Li. Sample-level data selection for federated learning. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021a.

L. Li, Y. Fan, M. Tse, and K.-Y. Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020a.

Q. Li, B. He, and D. Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021b.

Q. Li, Y. Diao, Q. Chen, and B. He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 965–978. IEEE, 2022.

T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020b.

X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

Z. Li, Y. Sun, J. Shao, Y. Mao, J. H. Wang, and J. Zhang. Feature matching data synthesis for non-iid federated learning. *IEEE Transactions on Mobile Computing*, 2024.

Y. Long, Z. Xue, L. Chu, T. Zhang, J. Wu, Y. Zang, and J. Du. Fedcd: A classifier debiased federated learning framework for non-iid data. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8994–9002, 2023.

M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34:5972–5984, 2021.

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017a.

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017b.

D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, and A. Y. Zomaya. Federated learning for covid-19 detection with generative adversarial networks in edge cloud computing. *IEEE Internet of Things Journal*, 9(12):10257–10271, 2021.

X. Pan, J. Yao, H. Kou, T. Wu, and C. Xiao. Harmonicnerf: Geometry-informed synthetic view augmentation for 3d scene reconstruction in driving scenarios. In *ACM Multimedia 2024*, 2023.

J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

E. Rizk, S. Vlaski, and A. H. Sayed. Optimal importance sampling for federated learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3095–3099. IEEE, 2021.

E. Rizk, S. Vlaski, and A. H. Sayed. Federated learning under importance sampling. *IEEE Transactions on Signal Processing*, 70:5381–5396, 2022.

C. Tian, L. Li, Z. Shi, J. Wang, and C. Xu. Harmony: Heterogeneity-aware hierarchical management for federated learning system. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 631–645. IEEE, 2022.

T. Tuor, S. Wang, B. J. Ko, C. Liu, and K. K. Leung. Data selection for federated learning with relevant and irrelevant data at clients. *arXiv preprint arXiv:2001.08300*, page 64, 2020.

S. Wan, J. Lu, P. Fan, Y. Shao, C. Peng, and K. B. Letaief. Convergence analysis and system design for federated learning over wireless networks. *IEEE Journal on Selected Areas in Communications*, 39(12):3622–3639, 2021.

H. Wang, Z. Kaplan, D. Niu, and B. Li. Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE INFOCOM 2020-IEEE conference on computer communications*, pages 1698–1707. IEEE, 2020.

J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor. A novel framework for the analysis and design of heterogeneous federated learning. *IEEE Transactions on Signal Processing*, 69:5234–5249, 2021.

X. Wu, H. Huang, Y. Ding, H. Wang, Y. Wang, and Q. Xu. Fednp: Towards non-iid federated learning via federated neural propagation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10399–10407, 2023.

C. Xiao and Y. Liu. A multifrequency data fusion deep learning model for carbon price prediction. *Journal of Forecasting*, 2024.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

L. Yang, J. Huang, W. Lin, and J. Cao. Personalized federated learning on non-iid data via group-based meta-learning. *ACM Transactions on Knowledge Discovery from Data*, 17(4):1–20, 2023.

J. Yao, Y. Lai, H. Kou, T. Wu, and R. Liu. Qe-bev: Query evolution for bird's eye view object detection in varied contexts. In *ACM Multimedia 2024*, 2024.

M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.

K. Yin, Z. Ding, Z. Dong, D. Chen, J. Fu, X. Ji, G. Yin, and Z. Wang. Nipd: A federated learning person detection benchmark based on real-world non-iid data. *arXiv preprint arXiv:2306.15932*, 2023.

M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pages 7252–7261. PMLR, 2019.

L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10174–10183, 2022.

W. Zhang, X. Wang, P. Zhou, W. Wu, and X. Zhang. Client selection for federated learning with non-iid data in mobile edge computing. *IEEE Access*, 9:24462–24474, 2021.

Z. Zhang, X. Hu, J. Zhang, Y. Zhang, H. Wang, L. Qu, and Z. Xu. Fedlegal: The first real-world federated learning benchmark for legal nlp. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3492–3507, 2023.

Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

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

Z. Zhu, Y. Shi, P. Fan, C. Peng, and K. B. Letaief. Isfl: Federated learning for non-iid data with local importance sampling. *IEEE Internet of Things Journal*, 2024.

A Appendix

A.1 Supplementary Explanation of the Diffusion-based Data Harmonization Mechanism

Forward Diffusion Process In the forward diffusion process, Gaussian noise with variance $\beta_t \in (0, 1)$ is added gradually to the sample $\mathbf{x}_0$ for T steps. The process is defined as:

$$q(\mathbf{z}_{1:T}|z_i) = \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{z}_{t-1}), \tag{16}$$

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t\mathbf{I}). \tag{17}$$

Using notations $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, the sample $\mathbf{z}_t$ can be defined directly as:

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t} z_i + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}). \tag{18}$$

Reverse Denoising Process The reverse denoising process aims to sample reversely from $\mathbf{z}_T$ through transition probabilities $q(\mathbf{z}_{t-1}|\mathbf{z}_t)$ for timesteps $T-1$ through 1 to obtain a sample drawn from $q(z_i)$. The transition $q(\mathbf{z}_{t-1}|\mathbf{z}_t)$ is a Gaussian distribution, tractable when conditioned on z_i:

$$q(\mathbf{z}_{t-1}|\mathbf{z}_t, z_i) = \mathcal{N}(\mathbf{z}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{z}_t, z_i), \tilde{\beta}_t\mathbf{I}), \tag{19}$$

where the mean $\tilde{\boldsymbol{\mu}}_t$ and variance $\tilde{\beta}_t$ are calculated as:

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{z}_t, z_i) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\right), \tag{20}$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{21}$$

The reverse transition probability $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t)$ relies on the entire data distribution and is approximated through a neural network:

$$p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t) = N(\mathbf{z}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{z}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{z}_t, t)), \tag{22}$$

where $\boldsymbol{\Sigma}_\theta(\mathbf{z}_t, t) = \tilde{\beta}_t\mathbf{I}$ and the mean $\boldsymbol{\mu}_\theta(\mathbf{z}_t, t)$ depends on a noise sample $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)$ learned by a neural network. The learning process is guided by the objective function:

$$L = \mathbb{E}_{t, z_i, \epsilon}\left[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)||^2\right], \tag{23}$$

while the output sample is obtained as:

$$\mathbf{z}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)\right) + \sigma_t\mathbf{z}, \tag{24}$$

where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ if $t > 1$ and $\mathbf{z} = 0$ otherwise.

A.2 Proof of Theorem 3.1

Proof. For convenience, define $\omega_i = \omega_i(\theta_t) = \sigma_i^* + (l_i - \tau)\, k + 2\,\lambda\, \frac{\log \sigma_i^*}{\sigma_i^*}\, k$, ,which can be regarded as the effective weight term induced by the indicator function $I_\lambda(l_i, \sigma_i)$. It appears in the local update of client i. Hence, the update rule can be written as:

$$\theta_{t+1} \;=\; \theta_t \;-\; \eta\, \omega_i\, \nabla_\theta l_i. \tag{25}$$

Meanwhile, we assume the local loss function $l_i(\theta)$ satisfies two common assumptions:

1. **L-smoothness (Lipschitz continuity of gradients):** The gradient of $l_i(\theta)$ is L-Lipschitz, i.e.,
$$\|\nabla l_i(\theta_1) - \nabla l_i(\theta_2)\| \;\le\; L\,\|\theta_1 - \theta_2\| \quad \text{for all } \theta_1, \theta_2. \tag{26}$$

2. **Bounded below:** There exists a constant l_i^* such that
$$l_i(\theta) \;\ge\; l_i^* \tag{27}$$
for all θ.

From L-smoothness, we have for the update from iteration t to $t+1$:
$$l_i(\theta_{t+1}) \;\le\; l_i(\theta_t) \;+\; \big\langle \nabla l_i(\theta_t),\, \theta_{t+1} - \theta_t \big\rangle \;+\; \frac{L}{2}\|\theta_{t+1} - \theta_t\|^2. \tag{28}$$

Using (25), we get
$$\theta_{t+1} - \theta_t \;=\; -\eta\,\omega_i\,\nabla_\theta l_i(\theta_t). \tag{29}$$

Substituting back yields
$$\begin{aligned} l_i(\theta_{t+1}) \;&\le\; l_i(\theta_t) \;-\; \eta\,\omega_i\,\|\nabla l_i(\theta_t)\|^2 \;+\; \frac{L}{2}\,\eta^2\,\omega_i^2\,\|\nabla l_i(\theta_t)\|^2 \\ &=\; l_i(\theta_t) \;-\; \Big(\eta\,\omega_i \;-\; \frac{L}{2}\,\eta^2\,\omega_i^2\Big)\|\nabla l_i(\theta_t)\|^2. \end{aligned} \tag{30}$$

If we ensure $\eta\,\omega_i \;\le\; \frac{1}{L}$, or more conservatively $\eta\,\omega_i \;\le\; \frac{1}{2\,L}$,then
$$\eta\,\omega_i \;-\; \frac{L}{2}\,\eta^2\,\omega_i^2 \;\ge\; \frac{1}{2}\,\eta\,\omega_i. \tag{31}$$

Hence, (30) implies
$$l_i(\theta_{t+1}) \;\le\; l_i(\theta_t) \;-\; \frac{1}{2}\,\eta\,\omega_i\,\|\nabla l_i(\theta_t)\|^2. \tag{32}$$

Therefore, if η is chosen such that $\eta\,\omega_i \;\le\; \frac{1}{2\,L}$, then $l_i(\theta)$ is guaranteed to decrease at every step, forming a monotonically decreasing sequence.

By the bounded-below assumption $l_i(\theta) \ge l_i^*$, the loss cannot decrease indefinitely. Consequently, $\{\, l_i(\theta_t)\}$ converges. From a standard telescoping sum argument, at iteration T, we estimate the sum of squared gradient norms:
$$\sum_{t=0}^{T-1}\big\|\nabla l_i(\theta_t)\big\|^2 \;\le\; \frac{2}{\eta}\sum_{t=0}^{T-1}\frac{l_i(\theta_t) - l_i(\theta_{t+1})}{\omega_i} \;\le\; \frac{2\,\big[l_i(\theta_0) - l_i^*\big]}{\eta\,\min_t\{\omega_i(\theta_t)\}}. \tag{33}$$

If $\min_t\{\omega_i(\theta_t)\} > 0$ and η is properly chosen, then the average gradient norm goes to 0 as $T \to \infty$, ensuring global convergence.

In FedAvg, the single-step update typically has the form $\theta_{t+1} = \theta_t - \eta_{\text{FedAvg}}\,\nabla_\theta l_i(\theta_t)$. In CRFed, we have an additional factor $\omega_i(\theta_t)$. Thus the effective learning rate becomes $\eta_{\text{eff}} = \eta\,\omega_i(\theta_t)$. If $\omega_i(\theta_t) \le 1$ (which can be ensured by proper design of the indicator function in many scenarios), then $\eta_{\text{eff}} \le \eta$, i.e., CRFed is more conservative (or adaptive) in its step size. This helps avoid large updates caused by data heterogeneity and promotes more stable convergence. Consequently, CRFed obtains a tighter convergence bound because it mitigates the gradient-directional deviation brought on by heterogeneous data.

□

A.3 Proof of Theorem 3.2

Proof. Recall that the Indicator Function is defined as $I_\lambda(l_i, \sigma_i) \;=\; (l_i - \tau)\,\sigma_i \;+\; \lambda\,(\log \sigma_i)^2$., we wish to find the optimal σ_i^* that minimizes this function for a given l_i.

To simplify the differentiation, we introduce two transformations:

$$c_i = \frac{l_i - \tau}{\lambda} \quad \text{and} \quad x_i = \log \sigma_i. \tag{34}$$

Hence, we have

$$\sigma_i \;=\; e^{\,x_i}. \tag{35}$$

Under these new variables, the Indicator Function becomes

$$I_\lambda(l_i, \sigma_i) \;=\; (l_i - \tau)\,e^{\,x_i} \;+\; \lambda\,(x_i)^2 \;=\; \lambda\Big(c_i\,e^{\,x_i} \;+\; x_i^2\Big). \tag{36}$$

Since $\lambda > 0$ is just a constant multiplier, minimizing I_λ is equivalent to minimizing

$$f(x_i) \;=\; c_i\,e^{\,x_i} \;+\; x_i^2. \tag{37}$$

We now compute the derivative of $f(x_i)$ with respect to x_i and set it to zero to find the critical points:

$$\frac{\mathrm{d}}{\mathrm{d}x_i}\big[\,c_i\,e^{\,x_i} + x_i^2\big] \;=\; c_i\,e^{\,x_i} \;+\; 2\,x_i \;=\; 0. \tag{38}$$

Rearrange this to isolate exponential terms:

$$c_i\,e^{\,x_i} \;=\; -\,2\,x_i. \tag{39}$$

Note that this step implicitly assumes $x_i < 0$ if the right-hand side is negative, depending on the sign of c_i. We proceed to manipulate this into a standard Lambert W form. Multiply both sides by $-\frac{1}{2}\,e^{\,x_i}$:

$$-\tfrac{1}{2}\,c_i\,e^{\,2\,x_i} \;=\; x_i\,e^{\,x_i}. \tag{40}$$

Let

$$y \;=\; x_i\,e^{\,x_i}. \tag{41}$$

Then,

$$y \;=\; -\tfrac{1}{2}\,c_i\,e^{\,2\,x_i}. \tag{42}$$

Meanwhile, by definition of y,

$$y \;=\; x_i\,e^{\,x_i}. \tag{43}$$

Hence, we arrive at

$$x_i \;=\; -\,W\Big(\tfrac{c_i}{2}\Big) \quad\Longrightarrow\quad x_i \;=\; -\,W\Big(\tfrac{l_i - \tau}{2\,\lambda}\Big), \tag{44}$$

where $W(\cdot)$ is the Lambert W function, the inverse function of $z \mapsto z\,e^{\,z}$.

In practice, the argument of the Lambert W function must lie in a domain where the function is real-valued. By restricting

$$l_i - \tau \geq -\tfrac{2}{e}, \tag{45}$$

we ensure that

$$\frac{l_i - \tau}{2\,\lambda} \geq -\tfrac{1}{e}. \tag{46}$$

Therefore, to handle the case when $l_i - \tau < -\frac{2}{e}$, we take

$$\max\Big(-\tfrac{2}{e},\, l_i - \tau\Big), \tag{47}$$

which ensures the Lambert W functions argument stays within the valid real domain. Consequently, the solution for x_i becomes

$$x_i = -\,W\Big(\tfrac{1}{2\,\lambda}\,\max\big(-\tfrac{2}{e},\, l_i - \tau\big)\Big). \tag{48}$$

Recalling $\sigma_i = e^{\,x_i}$, we conclude:

$$\sigma_i^*(l_i) \;=\; \exp\Big(-\,W\big(\tfrac{1}{2\,\lambda}\,\max\big(-\tfrac{2}{e},\, l_i - \tau\big)\big)\Big). \tag{49}$$

□

A.4 Design and Training Details of the Model Encoder and Distribution Decoder

In the CRFed framework, both the model encoder and distribution decoder play crucial roles in ensuring effective data harmonization and robust model updates. These components are implemented using autoencoder architectures, designed to compress and reconstruct data representations efficiently.

A.4.1 Model Encoder

The model encoder E is responsible for compressing the global model parameters θ_t into a lower-dimensional meta-model representation ϕ_t. The encoder architecture comprises several fully connected layers activated by ReLU functions, followed by a linear transformation layer to produce the final compressed representation.

Mathematically, given the input global model parameters $\theta_t \in \mathbb{R}^d$, the encoder outputs a compressed representation $\phi_t = E(\theta_t) \in \mathbb{R}^{d'}$, where $d' < d$. The transformation is defined as follows:

$$h_1 = \text{ReLU}(W_1\theta_t + b_1)h_2 = \text{ReLU}(W_2h_1 + b_2)\vdots h_k = \text{ReLU}(W_kh_{k-1} + b_k)\phi_t = W_{out}h_k + b_{out} \tag{50}$$

The training of the model encoder involves minimizing the mean squared error (MSE) between the original model parameters and their reconstructions. The loss function is given by:

$$\mathcal{L}_E = \frac{1}{N}\sum_{i=1}^{N} \|\theta_{t_i} - \hat{\theta}_{t_i}\|^2 \tag{51}$$

where $\hat{\theta}_{t_i} = E^{-1}(E(\theta_{t_i}))$ and N is the number of samples. The optimization is performed using the Adam optimizer with a learning rate η. The detailed architecture includes an input layer of size d, hidden layers of sizes $[128, 64, 32]$, and an output layer of size $d' = 16$.

A.4.2 Distribution Decoder

The distribution decoder D aims to transform the denoised latent representations $\mathbf{z}_t$ back into the desired data distribution. Like the encoder, the decoder uses a series of fully connected layers with ReLU activations, culminating in a linear layer to reconstruct the data.

Given the input latent representation $\mathbf{z}_t \in \mathbb{R}^{d'}$, the decoder outputs the reconstructed data $\hat{\mathbf{x}}_t = D(\mathbf{z}_t) \in \mathbb{R}^d$. The transformations are defined as:

$$h_1 = \text{ReLU}(W_1'\mathbf{z}_t + b_1')h_2 = \text{ReLU}(W_2'h_1 + b_2')\vdots h_k = \text{ReLU}(W_k'h_{k-1} + b_k')\hat{\mathbf{x}}_t = W_{out}'h_k + b_{out}' \tag{52}$$

The training process for the distribution decoder also minimizes the MSE, defined as:

$$\mathcal{L}_D = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_{t_i} - \hat{\mathbf{x}}_{t_i}\|^2 \tag{53}$$

where $\hat{\mathbf{x}}_{t_i} = D(\mathbf{z}_{t_i})$ and N is the number of samples. The optimization employs the Adam optimizer with a learning rate η. The architecture details include an input layer of size $d' = 16$, hidden layers of sizes $[32, 64, 128]$, and an output layer of size d.

A.4.3 Architectural Details

The model encoder and distribution decoder share a similar architectural approach, emphasizing efficient compression and reconstruction through deep learning techniques. The key parameters for both autoencoders are summarized as follows:

- **Model Encoder:**
  - Input layer size: d
  - Hidden layer sizes: $[128, 64, 32]$
  - Output layer size: $d' = 16$

- Activation function: ReLU
- Learning rate: $\eta = 0.001$
- Batch size: 32

- **Distribution Decoder:**
  - Input layer size: $d' = 16$
  - Hidden layer sizes: $[32, 64, 128]$
  - Output layer size: d
  - Activation function: ReLU
  - Learning rate: $\eta = 0.001$
  - Batch size: 32

These architectural and training details ensure that the CRFed framework can effectively handle non-i.i.d. data distributions, facilitating robust and consistent model updates across federated learning environments.

A.5 Pseudocode for CRFed

The complete computational process of CRFed is illustrated in Algorithm 1.

Algorithm 1 Confusion-Resistant Federated Learning via Consistent Diffusion (CRFed)

Require: Maximum global rounds T_G, local training cycles E_l, client weights $\{\pi_k\}$, local datasets $\{\mathcal{D}_k\}$, learning rate η_0, indicator function threshold γ

Initialize global model parameters θ_0
Initialize local model parameters $\{\theta_0^k\}$ and importance sampling weights $\{w_i^k \leftarrow 1\}$
Set $t \leftarrow 1$
while $t \leq T_G \times E_l$ **do**
  for each client k **do**
    Sample local data $\mathcal{D}_i^{\text{sampled}}$ based on importance weights w_i^k
    Train local model θ_t^k on $\mathcal{D}_i^{\text{sampled}}$
  end for
  if $t \mod E_l == 0$ **then**
    Each client uploads local model $\{\theta_t^k\}$ to the server
    Server aggregates the global model: $\bar{\theta}_t \leftarrow \sum_{k=1}^{K} \pi_k \theta_t^k$
    for each client k **do**
      Compute optimal indicator function I^*
      Calculate optimal sampling probability $P_i^* = \text{ReverseDenoise}(I^*)$
      Calculate importance sampling weights $w_i = \frac{P_i^*}{P_0}$
      Sample new local data $\mathcal{D}_i^{\text{sampled}}$ based on updated importance weights w_i
      Train local model θ_t^k on $\mathcal{D}_i^{\text{sampled}}$
      Update local model $\theta_t^k \leftarrow \bar{\theta}_t$
      Adjust learning rate: $\eta_i = \eta_0 \cdot \frac{I_\lambda(l_i, \sigma_i)}{\max_j I_\lambda(l_j, \sigma_j)}$
    end for
  end if
  $t \leftarrow t + 1$
end while
Output: Global model $\bar{\theta}_t$, local models $\{\theta_t^k\}$

A.6 Detailed Model Structure

The detailed configuration of the models used in our experiments is provided below. Each table outlines the layers and parameters for the respective datasets.

All weights are initialized with a normal distribution (mean 0, standard deviation 0.1) and biases with a constant value of 0.1. These settings ensure that the models are well-prepared for training and capable of achieving high performance on the respective datasets.

Table 4: Model structure for MNIST and FashionMNIST datasets

Layer	Type	Output Channels/Units	Additional Information
Input	-	-	28x28 grayscale images
1	Convolutional	16	5x5 filters, stride 1, padding 'SAME'
-	Activation	-	ReLU
-	Max Pooling	-	2x2 window, stride 2
2	Convolutional	32	5x5 filters, stride 1, padding 'SAME'
-	Activation	-	ReLU
-	Max Pooling	-	2x2 window, stride 2
3	Fully Connected	512	-
-	Activation	-	ReLU
4	Fully Connected	10	Softmax

Table 5: Model structure for CIFAR-10 and CIFAR-100 datasets

Layer	Type	Output Channels/Units	Additional Information
Input	-	-	32x32 RGB images
1	Convolutional	64	5x5 filters, stride 1, padding 'SAME'
-	Activation	-	ReLU
-	Max Pooling	-	2x2 window, stride 2
2	Convolutional	64	5x5 filters, stride 1, padding 'SAME'
-	Activation	-	ReLU
-	Max Pooling	-	2x2 window, stride 2
3	Fully Connected	1600	-
-	Activation	-	ReLU
4	Fully Connected	512	-
-	Activation	-	ReLU
5	Fully Connected	10 (CIFAR-10) / 100 (CIFAR-100)	Softmax

In the NIPD dataset, we adopted the classic YOLOv3 [Redmon and Farhadi, 2018] model.

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: The main claims in the abstract and introduction are supported by detailed descriptions, empirical evaluations, and theoretical analysis provided in the body of the paper (Sections 1, 3, and 4).

   Guidelines:

   - The answer NA 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 NA 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: The paper discusses the limitation in Section 5, addressing potential biases from large language models.

   Guidelines:

   - The answer NA 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: [Yes]

Justification: The paper includes a detailed proof for the convergence theorem in Section 3, providing all necessary assumptions and a complete proof.

Guidelines:

- The answer NA 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: The paper specifies the datasets used, the training and testing details, the hyperparameters, and the evaluation metrics in Section 4, ensuring that the experiments can be reproduced.

Guidelines:

- The answer NA 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: The data and code is provided.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides comprehensive details on the training and testing setups, including data splits, hyperparameters, and optimizer settings in Section 4.

   Guidelines:

   - The answer NA 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: Since some baselines involve randomness using k-means, the paper did 95% significance test using 10 repeated results.

   Guidelines:

   - The answer NA 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: The paper provides details on the computational resources used, including the type of GPUs and the total amount of compute required for the experiments, as mentioned in Section 4.

   Guidelines:

   - The answer NA 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: The research adheres to the NeurIPS Code of Ethics, ensuring transparency, reproducibility, and consideration of ethical implications throughout the study.

   Guidelines:

   - The answer NA 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 paper discusses the potential positive impacts of improving clustering techniques for various applications and mentions possible negative impacts such as biases introduced by large language models and the risk of misuse in surveillance, along with mitigation strategies in Section 5.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not involve the release of data or models that have a high risk for misuse.

    Guidelines:

    - The answer NA 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: The paper properly credits the creators of the datasets and models used, and mentions the licenses and terms of use, as detailed in Section 4.

    Guidelines:

    - The answer NA 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: The new datasets and code introduced in the paper are well documented.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA 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.