FORGETTING WITHOUT SEEING: BLIND FEDERATED UNLEARNING WITH PURE RANDOM NOISE

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated unlearning aims to remove the influence of a designated client from a trained global model while preserving model utility on the retained clients. Existing methods presuppose a fixed degree of unlearning client participation, whereas client availability is unpredictable in practice. Therefore, we formalize this setting as blind federated unlearning, where the unlearning client may never participate or may depart at any round, and the remaining rounds are completed solely by the retained clients. We find that when the unlearning client departs after partial participation, continuing with retained clients alone can substantially reverse the unlearning already achieved, revealing the need for persistent unlearning signals after client departure. To provide persistent surrogate signals without relying on semantic information about the unlearning data or auxiliary data, we propose Noise-as-Proxy Erasure (NoPE), which uses pure random noise generated at the server as proxy data. NoPE employs distance-based loss weighting to regulate the contribution of individual noise samples, enabling fine-grained control over the unlearning process. Extensive experiments demonstrate that NoPE achieves effective unlearning while maintaining competitive performance on retained data across different blind federated unlearning scenarios, and can be attached to existing federated unlearning methods to continue unlearning after client departure.

1 INTRODUCTION

With growing privacy concerns and data deletion requirements, removing client-specific influence has become an important capability for federated learning (McMahan et al., 2017; Yang et al., 2019). Retraining the global model after excluding the unlearning client provides a straightforward solution but incurs prohibitive computational and communication costs. Therefore, Federated Unlearning (FU) (Wu et al., 2022b; Liu et al., 2023) seeks to efficiently remove the influence of an unlearning client from a trained global model while preserving the performance on retained data.

Existing FU methods approximate the removal of the contribution of an unlearning client through gradient ascent (Liu et al., 2021a; Hua et al., 2024), model parameter correction (Wang et al., 2025; Wu et al., 2026), or knowledge distillation (Zhang et al., 2025a; Mora et al., 2026b). Nevertheless, many of those methods rely on the unlearning client remaining available throughout the unlearning phase to provide model updates, data statistics, or other information derived from the unlearning data, all of which serve as direct signals for progressively removing the target influence. Recent work (Mora et al., 2024; 2026a) reduces this dependence to single-round participation of the unlearning client but still requires the unlearning client to participate when needed. However, in practical federated systems, sustained participation cannot always be guaranteed due to network instability, hardware disconnection, or voluntary departure, and even single-round participation may be unavailable. Meanwhile, another line of unlearning-client independent FU enables unlearning without client participation throughout the entire unlearning phase via historical update reconstruction (Liu et al., 2021a; Huynh et al., 2024), model perturbation (Khalil et al., 2025), or previously obtained client information (Fu et al., 2024). Therefore, these methods are designed around different assumptions on unlearning client availability, but do not explicitly address unpredictable changes in availability during the unlearning process.

In this work, we consider a more general setting without assuming how many unlearning rounds the unlearning client can participate in. We formalize this setting as **blind federated unlearning**, where

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

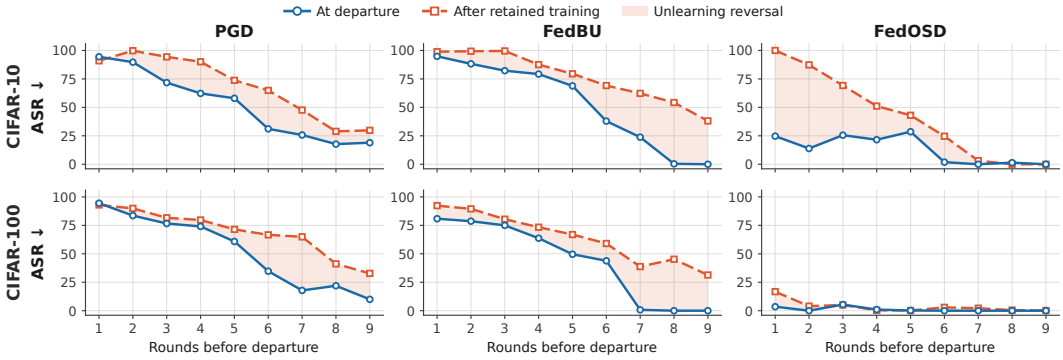


Figure 1: Dependence of existing federated unlearning methods on continued unlearning client participation. We vary the departure round within 10 unlearning rounds, with retained clients completing the remainder, and evaluate unlearning using attack success rate (ASR).

the unlearning client may never participate or may depart at any round, after which the remaining rounds are completed solely by retained clients. When the client is absent from the outset, which is the setting of unlearning-client independent methods, unlearning starts from the trained global model that still carries its full influence. When the client departs after partial participation, the remaining rounds instead continue from an intermediate model from which this influence has been only partially removed. This raises a question: whether the retained clients alone can complete the remaining unlearning starting from such an intermediate model.

To investigate this question, we fix the overall unlearning schedule while varying the duration of unlearning client participation, and compare the model at client departure with that after the retained clients complete the remaining rounds, with experimental details provided in Appendix B.4. As shown in Figure 1, the earlier the unlearning client departs, the more of its influence remains in the model at departure, as reflected by a higher ASR. More importantly, once the unlearning client departs, continued training with retained clients does not reliably preserve the unlearning already achieved and can even substantially reverse it. This finding indicates that retained-client training alone cannot substitute for the missing direct unlearning supervision, motivating the need for persistent surrogate signals after the unlearning client departure. Proxy data provides a natural solution for supplying such signals without continued participation of the unlearning client. However, relying on auxiliary or semantically related proxy data requires additional data resources or prior knowledge about the unlearning data. Such requirements may be impractical in federated learning settings and may also raise privacy concerns. This motivates a key question: *can we forget without seeing the unlearning data or semantically related external proxy data*?

To answer this question, we propose Noise-as-Proxy Erasure (NoPE), which uses pure random noise generated at the server as proxy data to construct persistent surrogate unlearning signals without requiring auxiliary data or semantic knowledge of the unlearning data. Specifically, NoPE employs distance-based loss weighting that assigns larger weights to noise samples closer to their class-specific sampling centers, which improves the utility preservation without weakening the unlearning. Since the noise is generated only once at the server with little overhead, NoPE can continuously provide surrogate unlearning signals after the unlearning client becomes unavailable, naturally accommodating both complete absence and partial participation in blind federated unlearning.

Our work makes three main contributions. First, we formulate blind federated unlearning without assuming a fixed degree of unlearning client participation, and reveal a failure mode where continued training with retained clients after unlearning client departure can reverse the unlearning already achieved. Second, we propose NoPE, which uses pure random noise with distance-based loss weighting to provide surrogate unlearning signals without requiring auxiliary data or prior knowledge of the unlearning data. Finally, extensive experiments demonstrate that NoPE achieves effective unlearning while maintaining strong performance on retained data across diverse blind federated unlearning settings and can serve as a plug-in for existing unlearning-client dependent methods by providing surrogate signals after the client departure.

2 Related Work

Federated Unlearning. Existing FU methods can be organized by how much participation they require from the unlearning client. The first line of research removes the influence of an unlearning client through gradient ascent (Hua et al., 2024; Liu et al., 2024), model parameter correction (Kundalwal et al., 2026; Zeng et al., 2026), or knowledge distillation (Wu et al., 2022a; Su & Li, 2023), but these mothods rely on the unlearning client to continuously provide model updates, data statistics, or other information derived from the unlearning data as direct unlearning signals. More recent work has sought to reduce the participation required from the unlearning client (Mora et al., 2024; Nan et al., 2026). PUF (Mora et al., 2026a) further requires only a single round of unlearning client participation, where the server negates and rescales the resulting pseudo-gradient, but the unlearning signal still originates from the unlearning client itself. Another line of unlearning-client independent FU completely avoids unlearning client participation through historical update reconstruction (Zhang et al., 2023; Huynh et al., 2024), model perturbation (Zhang et al., 2023; Fraboni et al., 2024), or previously obtained client information (Fu et al., 2024; Zhao et al., 2025). Representative approaches include FedEraser (Liu et al., 2021a), which leverages historical updates for model reconstruction and influence removal, and NoT (Khalil et al., 2025), which combines weight negation with retained-data recovery. In contrast to assuming a fixed degree of unlearning client participation, our blind federated unlearning formulation accommodates varying client availability, ranging from complete absence to departure after partial participation. Therefore, existing unlearning-client independent FU represents the special case with no unlearning client participation, whereas blind federated unlearning further captures the signal discontinuity induced by partial participation, where unlearning must continue from a partially unlearned model.

Proxy Data in Machine Learning. Proxy data have been widely used as a substitute to support optimization when the original training data are unavailable, including in knowledge distillation (Le et al., 2024; Sun et al., 2025), source-free domain adaptation (Liu et al., 2021b; He et al., 2024), and data-free domain generalization (Jurjo et al., 2021; Wu et al., 2022a; Yao et al., 2025). In federated learning, proxy data have also been used to transfer knowledge under heterogeneous data or models (Luo et al., 2021; Tang et al., 2022). These methods generally assume that proxy data must be sufficiently aligned with the original task in semantics and distribution so as to provide useful signals (Li et al., 2022; Zhang et al., 2025b). However, obtaining auxiliary data that match the unlearning client distribution introduces additional assumptions on data availability, while constructing proxy data with generative models incurs additional computational overhead and prior knowledge requirements and may raise privacy concerns in federated learning. In contrast, our experimental results show that effective unlearning does not require proxy data to carry semantic information about the unlearning data. Based on this observation, NoPE pushes proxy data to their simplest form by using pure random noise as the proxy data under blind federated unlearning, thereby avoiding the cost of collecting in-distribution auxiliary data or training generative models.

3 Preliminaries and Problem Formulation

Federated Learning. We consider a federated learning system with K clients indexed by $k \in [K] \triangleq \{1, \dots, K\}$, where each client k holds a private dataset $\mathcal{D}_k$. The goal is to learn a global model θ^*_{FL} that minimizes the weighted sum of local objectives, $\theta^*_{\mathrm{FL}} = \arg\min_\theta \sum_{k=1}^K p_k \mathcal{L}(\mathcal{D}_k; \theta)$, where $p_k \geq 0$ denotes the aggregation weight of client k and satisfies $\sum_{k=1}^K p_k = 1$. At communication round t, each participating client starts from the current global model θ^t_{FL} and performs local training to obtain $\theta^{t+1}_{\mathrm{FL},k}$. The server then aggregates the local updates (McMahan et al., 2017) as $\theta^{t+1}_{\mathrm{FL}} = \theta^t_{\mathrm{FL}} - \eta_g \sum_{k=1}^K p_k \left(\theta^t_{\mathrm{FL}} - \theta^{t+1}_{\mathrm{FL},k}\right)$, where η_g denotes the global learning rate.

Federated Unlearning. After T_{FL} rounds of federated learning, we denote the resulting global model by $\theta^0 = \theta^{T_{\mathrm{FL}}}_{\mathrm{FL}}$, which serves as the pre-unlearning model. Given an unlearning client $u \in [K]$, federated unlearning aims to remove the influence of client u from θ^0 while preserving the model utility on the retained clients $R = [K] \setminus \{u\}$. An unlearning request from client u then initiates an unlearning phase of T rounds. In each unlearning round, retained clients continue local training on their private data, while many existing methods depend on the unlearning client to provide model updates, data statistics, or other information derived from the unlearning data as direct signals for

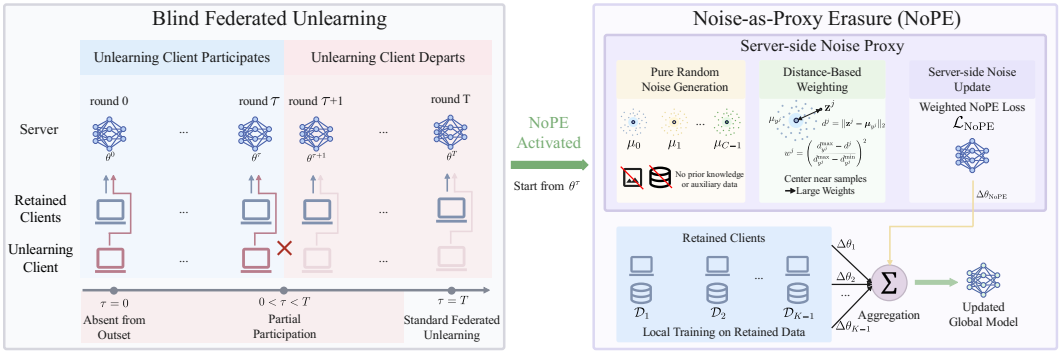


Figure 2: Overview of Blind Federated Unlearning and NoPE. When the unlearning client becomes unavailable, NoPE generates server-side pure random noise to provide surrogate unlearning signals while retained clients continue training.

removing its influence. In contrast, unlearning-client independent federated unlearning considers the case where the unlearning client remains unavailable throughout the entire unlearning phase. In this work, we assume a trusted server that faithfully executes the aggregation and unlearning protocols, while each client k keeps its private dataset $\mathcal{D}_k$ on its own device throughout training and unlearning.

Blind Federated Unlearning. Blind federated unlearning assumes neither access to the local unlearning data nor prior knowledge of how many unlearning rounds the unlearning client will remain available. Let $\tau \in \{0, \ldots, T\}$ denote the number of rounds the unlearning client u actually participates in before departure, and let θ^{τ} denote the global model at its departure. We define blind federated unlearning by $\tau < T$, where the unlearning client participates in the first τ rounds and does not return thereafter, leaving the remaining $T - \tau$ rounds to the retained clients. When $\tau = 0$, the unlearning client is absent from the outset, encompassing the unlearning-client independent federated unlearning setting. When $0 < \tau < T$, the unlearning client departs after partial participation, and the remaining unlearning process continues from θ^{τ}, where part of its influence has already been removed. For completeness, $\tau = T$ corresponds to the fully available federated unlearning setting and falls outside blind federated unlearning.

4 Methodology

Since τ is unknown in advance, blind federated unlearning requires effective surrogate signals whenever the unlearning client becomes unavailable. Therefore, we propose NoPE, which constructs a proxy from pure random noise at the server and exploits its sampling structure for distance-based loss weighting. Once the unlearning client becomes unavailable, NoPE uses the noise proxy to provide surrogate unlearning signals throughout the remaining unlearning rounds. Figure 2 provides an overview of the NoPE pipeline under blind federated unlearning.

4.1 Surrogate Signals from Random Noise

Once the unlearning client becomes unavailable, the server requires a proxy to construct surrogate unlearning signals without access to its local data. Since proxy data need only overwrite the influence of the unlearning client rather than reconstruct the representation of unlearning data, NoPE pushes proxy data to their simplest form by using pure random noise, which requires no auxiliary data aligned with the unlearning data and incurs no training cost for generative models.

Let $\mathcal{D}_u$ denote the local dataset of the unlearning client u and let $N = |\mathcal{D}_u|$. For a C-class classification task, NoPE constructs a server-side noise proxy dataset $\mathcal{Z} = \{(\mathbf{z}^j, y^j)\}_{j=1}^{N}$ with $|\mathcal{Z}| = |\mathcal{D}_u|$, allowing the proxy update to inherit the FedAvg aggregation weight of client u. Apart from this size, which the server already holds through the aggregation protocol, NoPE generates noise data $\mathbf{z}^j \in \mathcal{Z}$ at the server without semantic or distributional information from $\mathcal{D}_u$.

Let $\mathcal{X}$ denote the space in which the noise is generated. For the image modality, $\mathcal{X}$ is the input space of the model, whereas for the text modality, whose inputs are discrete tokens on which continuous

noise cannot be defined directly, $\mathcal{X}$ is the embedding space of the model. The dimension of $\mathcal{X}$ for each dataset is specified in Appendix B.2. Specifically, for each noise sample, we first sample a pseudo-label y^j uniformly from the class set $\mathcal{C} = \{0, \cdots, C-1\}$ and then draw $\mathbf{z}^j$ from the corresponding Gaussian distribution of that class,

$$y^j \sim \mathrm{Uniform}(\mathcal{C}), \qquad \mathbf{z}^j \sim \mathcal{N}\left(\boldsymbol{\mu}_{y^j}, \mathrm{diag}(\boldsymbol{\sigma}^2)\right), \tag{1}$$

where the class mean is $\boldsymbol{\mu}_c = c\,\boldsymbol{m}$ for a predefined base mean $\boldsymbol{m} \in \mathcal{X}$ and the standard deviation $\boldsymbol{\sigma}$ is shared across all classes. The class-specific means separate the noise distributions of different labels, while the shared standard deviation keeps the dispersion consistent across classes. Repeating this procedure N times yields the full proxy dataset $\mathcal{Z}$. In this way, $\mathcal{Z}$ takes the place of the inaccessible $\mathcal{D}_u$ and supplies the signal for removing the influence of client u.

4.2 Distance-Based Weighting of Noise Samples

Although the noise samples are generated randomly, samples sharing the same pseudo-label are drawn from the same Gaussian component and vary in their deviation from the corresponding class center. Since all classes share the same standard deviation, these distances are comparable across classes and provide a common reference for distance-based weighting. Therefore, NoPE uses the distance to the corresponding class center to regulate the contribution of each proxy sample.

For proxy sample $(\mathbf{z}^j, y^j) \in \mathcal{Z}$, we measure its Euclidean distance to the sampling center of its pseudo-label as $d^j = \|\mathbf{z}^j - \boldsymbol{\mu}_{y^j}\|_2$, and let $d_c^{\max}$ and $d_c^{\min}$ denote the maximum and minimum of the distances over the samples with pseudo-label $c \in \mathcal{C}$. NoPE then assigns proxy sample the weight

$$w^j = \left(\frac{d_{y^j}^{\max} - d^j}{d_{y^j}^{\max} - d_{y^j}^{\min}}\right)^2. \tag{2}$$

The weight lies in $[0, 1]$ and decreases monotonically as a sample moves away from its class center. Therefore, samples near the center, which represent the structure of the corresponding noise component more consistently, receive larger weights, whereas samples in the distribution tail, which exhibit larger random deviations, are down-weighted. The square further suppresses these peripheral samples and thereby reduces their contribution to the server-side update. Since $\mathcal{Z}$ remains fixed throughout the unlearning phase, all distances and weights are computed once at generation time and incur no additional cost during training.

Using these weights, NoPE optimizes the weighted cross-entropy objective over the noise proxy set

$$\mathcal{L}_{\mathrm{NoPE}}(\mathcal{Z}; \theta) = \frac{1}{|\mathcal{Z}|} \sum_{j=1}^{|\mathcal{Z}|} w^j\, \mathcal{L}_{\mathrm{CE}}\left(\mathbf{z}^j, y^j; \theta\right). \tag{3}$$

4.3 NoPE under Blind Federated Unlearning

NoPE is activated whenever the unlearning client becomes unavailable, either from the beginning of unlearning or after departure during the unlearning phase. When the unlearning client is absent from the outset, NoPE performs unlearning from the pre-unlearning model θ^0. When the unlearning client departs after partial participation, NoPE continues the remaining unlearning rounds from the departure model θ^τ. Upon activation, the server constructs the class-separable noise proxy $\mathcal{Z}$ and computes its distance-based weights. The proxy and its weights remain fixed throughout the subsequent unlearning rounds.

In each unlearning round, retained clients perform standard local training on their private data, while the server initializes a proxy model from the current global model and optimizes it over $\mathcal{Z}$ using $\mathcal{L}_{\mathrm{NoPE}}$ for the same number of optimization steps as the retained clients. The resulting proxy update is then aggregated with the retained-client updates through FedAvg. Therefore, NoPE supplements the missing unlearning signal at the server while leaving retained-client training and the standard aggregation procedure unchanged, and as a single procedure covers every case of blind federated unlearning. Moreover, its additional cost per round is equivalent to the local training of a single client and requires no extra communication. The complete procedure is provided in Appendix A.

Table 1: Unlearning performance under the unlearning client availability requirement of each method. Best results among unlearning methods are in **bold**.

	CIFAR-10				CIFAR-100			
	IID		Non-IID		IID		Non-IID	
Method	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓
REFERENCE								
Pre-Unlearning	81.24±1.6	99.71	62.53±1.0	99.80	49.29±1.0	99.15	41.35±5.0	97.51
Retraining	80.92±0.3	1.67	71.24±0.7	1.33	52.92±0.5	0.00	50.08±0.1	0.00
Retained-Client Training	81.03±1.2	100.00	63.02±0.8	99.62	49.47±0.5	100.00	44.37±2.7	98.49
UNLEARNING-CLIENT DEPENDENT								
PGD	30.63±2.0	4.60	28.94±1.1	16.25	38.93±0.5	**0.00**	25.13±0.1	12.26
FedBU	59.42±0.9	**0.00**	21.45±0.1	**0.00**	42.63±0.2	11.17	20.14±1.3	**0.00**
FedOSD	71.72±0.1	**0.00**	46.69±1.3	**0.00**	47.22±0.9	**0.00**	47.51±1.4	**0.00**
UNLEARNING-CLIENT INDEPENDENT								
FedEraser	48.45±0.7	0.10	27.62±0.2	5.57	37.24±1.0	0.39	25.04±0.2	0.17
NoT	75.33±0.3	8.72	42.27±1.2	**0.00**	48.50±0.3	4.16	39.87±0.5	3.79
NoPE (**Ours**)	**78.33±0.1**	**0.00**	**68.69±0.3**	**0.00**	**52.72±0.3**	**0.00**	**49.59±0.2**	**0.00**

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets and Models. We evaluate our method on CIFAR-10 and CIFAR-100 using ResNet-18 under IID and Non-IID settings. In the IID setting, we randomly shuffle and uniformly split the dataset across clients. In the Non-IID setting, we sample the class proportions of each client from a Dirichlet distribution $\mathrm{Dir}(\alpha)$ with $\alpha = 0.1$. For each client, we further construct a local test set whose class distribution matches that of its training data.

Baselines. We compare NoPE with representative unlearning-client dependent and unlearning-client independent FU methods, and report Pre-Unlearning, Retraining, and Retained-Client Training as references. Pre-Unlearning denotes the original global model θ_0 before unlearning, Retraining trains the model from scratch using only the retained clients, and Retained-Client Training continues training on the retained clients alone throughout the unlearning phase. Unlearning-client dependent methods require continued participation of the unlearning client to provide direct unlearning signals. We consider PGD (Halimi et al., 2022), which maximizes the unlearning loss under an ℓ_2 constraint, FedBU (Alam et al., 2023), which bounds residual information while accounting for retained utility, and FedOSD (Pan et al., 2024), which updates the model along an orthogonal steepest descent direction. Unlearning-client independent methods perform unlearning without participation of the unlearning client during the unlearning phase. We consider FedEraser (Liu et al., 2021a), which reconstructs the model from historical updates, and NoT (Khalil et al., 2025), which negates the weights of selected layers and then fine-tunes the model on the retained clients.

Implementation Details. We first train the global model for 500 federated rounds with a learning rate of 0.005, followed by $T = 10$ rounds of unlearning. Each client uses SGD with a batch size of 32 and one local epoch per communication round. For NoPE, we set the RGB base mean to $m = [0.5, 0.3, 0.7]^\top$ and the shared standard deviation to $\sigma = [0.2, 0.3, 0.4]^\top$, with both spatially replicated over the input dimensions. We consider ten clients with full participation in each communication round and randomly designate one client as the unlearning client throughout the unlearning phase. We evaluate model utility using test accuracy on the retained clients and unlearning effectiveness using the attack success rate (ASR). For the backdoor evaluation, we inject the trigger into all samples of the unlearning client, yielding a Pre-Unlearning model with an ASR of nearly 100%. For each baseline, we perform a grid search to identify its optimal learning rate over the range $[10^{-5}, 10^{-2}]$, reporting the best performance achieved. All results are averaged over three random seeds, and further details are provided in Appendix B.

Table 2: Unlearning performance after early departure of the unlearning client. Within 10 unlearning rounds, the client departs after 1 or 5 rounds, with only retained clients participating thereafter. Best results among unlearning methods after departure are in **bold** within each block.

		CIFAR-10				CIFAR-100			
		$\tau = 1$		$\tau = 5$		$\tau = 1$		$\tau = 5$	
Before Departure	After Departure	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓
PGD	At departure	68.22±0.3	94.44	50.94±0.0	57.99	49.81±0.2	94.49	36.86±0.6	60.84
	+ Retained-Client Training	85.07±0.1	90.95	77.37±0.0	73.78	52.28±0.0	92.98	54.23±0.2	71.48
	+ FedEraser	60.71±0.5	1.08	54.16±0.8	**0.00**	37.92±0.0	**0.00**	32.81±0.4	**0.00**
	+ NoT	62.47±0.2	2.30	59.95±0.4	0.48	42.79±0.0	0.54	37.93±0.0	**0.00**
	+ NoPE (**Ours**)	**72.20±0.3**	**0.12**	**69.79±0.1**	**0.00**	**53.45±0.1**	**0.00**	**53.68±0.3**	0.28
FedBU	At departure	62.71±0.2	94.86	48.47±0.4	68.88	47.96±0.0	80.84	39.71±0.7	49.61
	+ Retained-Client Training	85.14±0.0	98.91	78.39±0.0	79.52	54.91±0.2	92.33	47.24±0.0	66.94
	+ FedEraser	63.87±0.0	1.18	54.92±0.1	0.50	35.65±0.6	1.17	30.97±0.8	**0.00**
	+ NoT	61.68±0.0	1.04	56.48±0.0	**0.00**	42.97±0.7	0.20	40.71±1.1	**0.00**
	+ NoPE (**Ours**)	**79.82±0.0**	**0.14**	**79.11±0.1**	1.99	**52.19±0.2**	**0.00**	**48.64±0.2**	**0.00**
FedOSD	At departure	70.81±0.4	24.72	57.98±0.0	28.61	50.04±0.8	3.52	45.96±1.3	0.19
	+ Retained-Client Training	80.34±0.0	100.00	70.75±0.6	43.04	54.20±0.0	16.73	49.04±0.3	0.18
	+ FedEraser	65.68±0.6	2.71	59.14±0.2	**0.00**	35.14±0.9	2.07	25.29±1.0	**0.00**
	+ NoT	64.76±0.0	2.03	59.05±0.4	2.45	47.38±0.1	0.34	40.11±0.0	**0.00**
	+ NoPE (**Ours**)	**80.39±0.0**	**0.00**	**79.58±0.0**	**0.00**	**53.73±0.3**	**0.00**	**54.71±0.1**	**0.00**

5.2 Main Unlearning Results

We first compare NoPE with existing federated unlearning methods under their respective requirements on unlearning client availability. For unlearning-client dependent baselines, we allow the unlearning client to participate in all unlearning rounds. For unlearning-client independent baselines and NoPE, we exclude the unlearning client from the entire unlearning phase.

As shown in Table 1, existing unlearning-client dependent methods substantially reduce ASR with full unlearning client participation, but their retained accuracy degrades across the evaluated settings. Among unlearning-client independent methods, FedEraser reduces ASR at the cost of substantial performance degradation. NoT better preserves model utility but leaves residual ASR in several settings. Retained-Client Training alone preserves model utility but leaves ASR close to the pre-unlearning level, showing that continued training on retained clients is insufficient for effective unlearning. Since it differs from NoPE only in the absence of the noise proxy, the gap in ASR between the two can be attributed directly to the surrogate signal provided by NoPE. In contrast, NoPE reduces ASR to 0% across all evaluated settings while achieving the highest retained accuracy among the methods that effectively remove the target influence, despite operating without unlearning client participation. Additional results on membership inference attacks, scalability to more clients, robustness under varying client join rates, and the text modality are provided in Appendix C.

5.3 NoPE as a Plug-In for Unlearning-Client Dependent Methods

We further consider the partial participation setting of blind federated unlearning, where $0 < \tau < T$. Specifically, we allow the unlearning client to participate in $\tau \in \{1, 5\}$ rounds using unlearning-client dependent methods, and report the model at the moment of departure as At Departure. Starting from the same departure model θ^τ, we then use Retained-Client Training, FedEraser, NoT, or NoPE to complete the remaining $T - \tau$ rounds. Further experimental details and results on the text modality are provided in Appendices B and C.

As shown in Figure 1 and Table 2, longer unlearning client participation generally yields stronger unlearning at departure. However, continuing with retained clients alone often increases ASR from its level at departure, reversing part of the unlearning already achieved. FedEraser and NoT substantially reduce ASR after client departure, showing that dedicated unlearning mechanisms can mitigate this unlearning reversal, but generally incur greater utility degradation. In contrast, NoPE directly continues from the departure models produced by different unlearning-client dependent methods and consistently restores near-complete unlearning across participation durations and datasets while preserving higher performance than FedEraser and NoT in most settings. Moreover, NoPE

Table 3: Unlearning performance with different proxy data on CIFAR-10. Best results among unlearning-data independent proxies are in **bold**.

		Proxy Only				**Federated with Retained Clients**			
		IID		Non-IID		IID		Non-IID	
Proxy Data	Requires	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓
Label-Shuffled $\mathcal{D}_u$	$\mathcal{D}_u$ samples	60.74±0.3	0.00	37.56±0.9	0.00	76.59±0.4	0.00	64.51±0.2	1.08
Class-Wise Gaussian	$\mathcal{D}_u$ statistics	57.37±0.4	0.00	40.27±0.0	0.00	77.20±0.8	3.09	69.07±0.0	1.85
External Data (SVHN)	Auxiliary data	**60.09±0.4**	2.19	41.82±0.2	0.52	76.92±0.2	1.43	63.93±0.1	**0.00**
Indivisible Noise	None	57.92±0.0	**0.00**	36.05±0.2	**0.00**	73.16±0.3	2.72	63.29±0.6	0.04
Divisible Noise (**Ours**)	None	59.69±0.1	**0.00**	**41.96±0.2**	**0.00**	**78.33±0.1**	**0.00**	**68.69±0.3**	**0.00**

Table 4: Ablation of distance-based weighting. Best results across weighting variants are in **bold**.

	Weighting Design		**CIFAR-10**				**CIFAR-100**			
			IID		Non-IID		IID		Non-IID	
Variant	Preference	Squared	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓
Uniform	None		71.83±0.1	0.85	53.87±0.0	0.00	41.93±0.1	0.00	40.48±0.2	0.00
Linear Center-Near	Near		74.57±0.4	**0.00**	60.86±0.2	**0.00**	44.89±0.1	**0.00**	32.02±1.0	**0.00**
Squared Center-Far	Far	✓	77.12±0.0	**0.00**	67.08±0.2	**0.00**	50.74±0.3	**0.00**	**50.03±0.4**	**0.00**
Squared Center-Near (**Ours**)	Near	✓	**78.33±0.1**	**0.00**	**68.69±0.3**	**0.00**	**52.72±0.3**	**0.00**	49.59±0.2	**0.00**

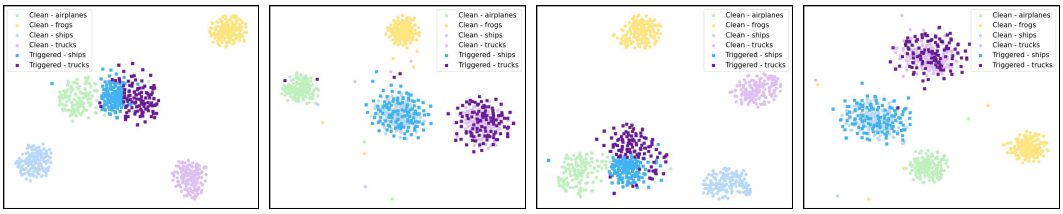
requires no modification to the unlearning-client dependent method before the unlearning client's departure, preserving the unlearning already achieved while removing the residual influence and thereby preventing unlearning reversal. These results demonstrate the potential of NoPE as a general post-departure plug-in for existing unlearning-client dependent federated unlearning methods.

5.4 Ablation Study

The Effect of Proxy Data. To understand how the choice of proxy data affects surrogate unlearning, we compare proxies from different sources on CIFAR-10, including label-shuffled unlearning data, a Gaussian proxy constructed from unlearning-data statistics, external auxiliary data, and random noise with and without class-separable structure, termed Divisible and Indivisible Noise. The first two are unavailable after the unlearning client departs and are included only as references that indicate what a server could reach if it retained access to the unlearning data or to their statistics. We evaluate each proxy under Proxy Only, which trains the pre-unlearning model solely on the proxy with the same training budget as NoPE, and Federated with Retained Clients, which substitutes the corresponding proxy into the NoPE process. Detailed proxy construction, additional results on the sensitivity of proxy size and overhead analysis of proxy data are provided in Appendices B, C and E.

As shown in Table 3, proxies from substantially different sources achieve strong unlearning under Proxy Only, suggesting that effective surrogate unlearning signals can arise without close semantic or statistical correspondence to the unlearning data. However, Proxy Only generally incurs substantial utility degradation, while collaboration between the server and the retained clients in the federated setting substantially recovers utility while largely preserving the unlearning achieved. This highlights the complementary effects of proxy data on unlearning and retained-client training on utility preservation. Notably, the proxies derived from the unlearning data yield no stronger unlearning than the noise proxy of NoPE and at most a marginal gain in utility, suggesting that access to the unlearning data is not necessary for effective unlearning. Among proxies independent of the unlearning data, Divisible Noise achieves better overall utility while maintaining nearly complete unlearning. Moreover, it can be generated through random sampling without accessing the unlearning data, introducing external data, or training additional generative models. Therefore, NoPE provides an effective surrogate signal at minimal construction cost under the data-availability constraints of blind federated unlearning.

The Effect of Distance-Based Weighting. To investigate the effect of distance-based weighting, we fix the noise proxy and all other training configurations while varying the weighting strategy. In addition to uniform weighting, we compare Linear Center-Near and Squared Center-Far weighting with the Squared Center-Near weighting adopted by NoPE. Further details are provided in Appendix B.



(a) Pre-Unlearning (b) Retraining (c) Retained-Client Training (d) NoPE (Ours)

Figure 3: t-SNE visualization of clean and triggered representations on CIFAR-10.

As shown in Table 4, uniform weighting already reduces ASR to near zero across all settings, showing that the noise proxy itself provides effective surrogate unlearning signals. However, its lower retained accuracy indicates that differentiating the contributions of individual noise samples is important for utility preservation. Distance-based weighting generally improves retained accuracy, with the squared center-near weighting adopted by NoPE achieving the highest accuracy in most settings and competitive performance in the remaining setting. It also achieves zero ASR across all settings. Compared with squared center-far weighting, emphasizing samples closer to their corresponding Gaussian centers generally provides more consistent utility preservation. These results support the weighting design of NoPE, which assigns larger weights to center-near samples while further suppressing samples farther from the centers.

5.5 Representation Analysis of Unlearning

To further examine how different unlearning methods affect the representations of backdoored samples, we use t-SNE to visualize the representations learned by Pre-Unlearning, Retraining, Retained-Client Training, and NoPE on the same set of clean and triggered samples. Following our backdoor setting as shown in Appendix B, triggered samples are reassigned during training to the target class 0, which corresponds to airplanes.

As shown in Figure 3, triggered samples under Pre-Unlearning deviate from their original classes and cluster near the clean representations of the target class. Retraining moves these samples away from the target-class cluster and closer to the clean representations of their original classes, providing a reference representation structure after effective unlearning. In contrast, under Retained-Client Training, the triggered samples remain close to the target-class cluster, indicating that continued training on retained data alone is insufficient to remove the learned backdoor representation. NoPE exhibits a representation shift similar to that of Retraining, moving triggered samples away from the target-class cluster and toward the clean clusters of their respective original classes. Since NoPE differs from Retained-Client Training only in the noise proxy, this contrast can be attributed directly to the surrogate signal it supplies. The visualization is consistent with the quantitative ASR results and further supports the effectiveness of NoPE from the perspective of learned representations.

6 Conclusion

In this work, we study Blind Federated Unlearning, where the unlearning client may be unavailable from the outset or depart during the unlearning phase. We find that when the client departs after partial participation, continuing with retained clients alone can reverse the unlearning already achieved. To address this challenge, we propose NoPE, which uses pure random noise to provide surrogate unlearning signals. The noise requires only one-time random sampling and no semantic information about the unlearning data or auxiliary data, allowing NoPE to provide surrogate signals whenever direct unlearning signals become unavailable. Extensive experiments demonstrate that NoPE achieves effective unlearning with competitive utility under blind federated unlearning, and that it can serve as a plug-in for unlearning-client dependent methods to mitigate unlearning reversal after client departure. Moreover, our analysis suggests that effective surrogate unlearning signals can arise without close semantic or statistical correspondence to the unlearning data, providing a new perspective on federated unlearning under limited client and data availability.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing and language polishing, including improving the wording, grammar, and clarity of parts of the manuscript. All AI-assisted text was manually reviewed, verified, and revised by the authors as necessary. The authors take full responsibility for the final content of this work, including its technical claims, experimental results, conclusions, and presentation.

ETHICS STATEMENT

This work aims to improve the feasibility of data removal in federated learning by enabling effective federated unlearning when the unlearning client cannot remain available throughout the unlearning process. NoPE constructs its proxy data from pure random noise at the server and does not require access to the semantic content of the unlearning data or additional external real data. All experiments are conducted on publicly available benchmark datasets and do not involve newly collected human-subject data or personally identifiable information.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide detailed methodological and experimental information throughout the main paper and appendix. The Methodology section describes the construction of the noise proxy, distance-based sample weighting, and the optimization procedure of NoPE under Blind Federated Unlearning, while the Experimental Setup specifies the main training configurations, federated learning settings, and evaluation metrics. Appendix A provides the complete algorithmic procedure of NoPE, and Appendix B further documents the datasets and models, noise construction across modalities, evaluation metrics, client departure and plug-in protocol, and the construction of proxy-data and weighting variants used in the ablation studies. The main experimental results are reported over multiple random seeds. We will also provide anonymized source code with the submission to further facilitate reproduction of our results.

REFERENCES

Manaar Alam, Hithem Lamri, and Michail Maniatakos. Get rid of your trail: Remotely erasing backdoors in federated learning. *CoRR*, abs/2304.10638, 2023.

Yann Fraboni, Martin Van Waerebeke, Kevin Scaman, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. SIFU: sequential informed federated unlearning for efficient and provable client unlearning in federated optimization. In *International Conference on Artificial Intelligence and Statistics*, volume 238, 2024.

Chaohao Fu, Weijia Jia, and Na Ruan. Client-free federated unlearning via training reconstruction with anchor subspace calibration. In *ICASSP*, 2024.

Anisa Halimi, Swanand Kadhe, Ambrish Rawat, and Nathalie Baracaldo. Federated unlearning: How to efficiently erase a client in fl? *CoRR*, abs/2207.05521, 2022.

Jiujun He, Liang Wu, Chaofan Tao, and Fengmao Lv. Source-free domain adaptation with unrestricted source hypothesis. *Pattern Recognit.*, 149, 2024.

Ying Hua, Hui Xia, and Shuo Xu. Federated unlearning for samples based on adaptive gradient ascent of angles. In *23rd IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2024.

Thanh Trung Huynh, Trong Bang Nguyen, Phi Le Nguyen, Thanh Tam Nguyen, Matthias Weidlich, Quoc Viet Hung Nguyen, and Karl Aberer. Fast-fedul: A training-free federated unlearning with provable skew resilience. In *ECML/PKDD*, 2024.

Manel Baradad Jurjo, Jonas Wulff, Tongzhou Wang, Phillip Isola, and Antonio Torralba. Learning to see by looking at noise. In *NeurIPS*, pp. 2556–2569, 2021.

Yasser H. Khalil, Leo Maxime Brunswic, Soufiane Lamghari, Xu Li, Mahdi Beitollahi, and Xi Chen. Not: Federated unlearning via weight negation. In *CVPR*, 2025.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014.

Mayank Kumar Kundalwal, Mamta, Deepak Mishra, and Asif Ekbal. Federated model synchronization for diagnostic redefinition through a novel selective parameter unlearning. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2026, Tucson, AZ, USA, March 6-10, 2026*, 2026.

Huy Q. Le, Minh N.H. Nguyen, Shashi Raj Pandey, Chaoning Zhang, and Choong Seon Hong. Cdkt-fl: Cross-device knowledge transfer using proxy dataset in federated learning. *Engineering Applications of Artificial Intelligence*, 133, 2024.

Zijian Li, Jiawei Shao, Yuyi Mao, Jessie Hui Wang, and Jun Zhang. Federated learning with gan-based data synthesis for non-iid clients. In *FL@IJCAI*, 2022.

Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. Federaser: Enabling efficient client-level data removal from federated learning models. In *IWQoS*, 2021a.

Ying Liu, Jialiang Peng, Sergey Kovalev, Igor Kotenko, Andrey Sukhanov, Yin Li, and Yao Li. Fedpcga: A federated unlearning method based on projected conflict gradient ascent. In *Intelligent Information Technologies for Industr*, volume 1209. Springer Nature Switzerland, 2024.

Zhe Liu, Ke Li, Shreyan Bakshi, and Fuchun Peng. Private language model adaptation for speech recognition. *CoRR*, abs/2110.10026, 2021b.

Ziyao Liu, Yu Jiang, Jiyuan Shen, Minyi Peng, Kwok-Yan Lam, and Xingliang Yuan. A survey on federated unlearning: Challenges, methods, and future directions. *CoRR*, abs/2310.20448, 2023.

Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. In *NeurIPS*, pp. 5972–5984, 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

Alessio Mora, Lorenzo Valerio, Paolo Bellavista, and Andrea Passarella. Fedquit: On-device federated unlearning via a quasi-competent virtual teacher. *CoRR*, abs/2408.07587, 2024.

Alessio Mora, Carlo Mazzocca, Rebecca Montanari, and Paolo Bellavista. Federated unlearning made practical: Seamless integration via negated pseudo-gradients. *IEEE Transactions on Big Data*, 12, 2026a.

Alessio Mora, Lorenzo Valerio, and Paolo Bellavista. Federated unlearning via distilled data. *IEEE Trans. Mob. Comput.*, 25, 2026b.

Feihong Nan, Zhengyi Zhong, Pan Wang, Weidong Bao, Xiongtao Zhang, Quan Wen, and Ji Wang. Fedup: One-shot federated unlearning via centroid-guided plug-in filters. In *IJCAI/ECAI*, 2026.

Zibin Pan, Zhichao Wang, Chi Li, Kaiyan Zheng, Boqi Wang, Xiaoying Tang, and Junhua Zhao. Federated unlearning with gradient descent and conflict mitigation. *CoRR*, abs/2412.20200, 2024.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, pp. 3–18, 2017.

Ningxin Su and Baochun Li. Asynchronous federated unlearning. In *INFOCOM*, 2023.

Chuanneng Sun, Tingcong Jiang, and Dario Pompili. Heterogeneous federated learning via generative model-aided knowledge distillation in the edge. *IEEE Internet Things J.*, 12, 2025.

Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xin He, Bo Han, and Xiaowen Chu. Virtual homogeneity learning: Defending against data heterogeneity in federated learning. In *ICML*, 2022.

Weiqi Wang, Chenhan Zhang, Zhiyi Tian, and Shui Yu. Fedu: Federated unlearning via user-side influence approximation forgetting. *IEEE Trans. Dependable Secur. Comput.*, 22, 2025.

Chen Wu, Sencun Zhu, and Prasenjit Mitra. Federated unlearning with knowledge distillation. *CoRR*, abs/2201.09441, 2022a.

Hengliang Wu, Youming Tao, Anhao Zhou, Shuzhen Chen, Falko Dressler, and Dongxiao Yu. Certified unlearning in decentralized federated learning. In *IEEE INFOCOM 2026 - IEEE Conference on Computer Communications, Tokyo, Japan, May 18-21, 2026*, 2026.

Leijie Wu, Song Guo, Junxiao Wang, Zicong Hong, Jie Zhang, and Yaohong Ding. Federated unlearning: Guarantee the right of clients to forget. *IEEE Netw.*, 36, 2022b.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10, 2019.

Yuan Yao, Xiaopu Zhang, Yu Zhang, Jian Jin, and Qiang Yang. Noise may contain transferable knowledge: Understanding semi-supervised heterogeneous domain adaptation from an empirical perspective. *CoRR*, abs/2502.13573, 2025.

Wenjun Zeng, Siguang Chen, Xue Li, and Songle Chen. FU-PA: federated unlearning via parameters adjustment. *IEEE Trans. Emerg. Top. Comput. Intell.*, 10, 2026.

Jianxin Zhang, Mengda Zhao, Zhenwei Wang, Weijian Su, and Pengfei Wang. Model recovery in federated unlearning with restricted server data resources. *IEEE Internet Things J.*, 12, 2025a.

Lefeng Zhang, Tianqing Zhu, Haibin Zhang, Ping Xiong, and Wanlei Zhou. Fedrecovery: Differentially private machine unlearning for federated learning frameworks. *IEEE Trans. Inf. Forensics Secur.*, 18, 2023.

Tianlong Zhang, Zhe Xue, Adnan Mahmood, Junping Du, Yuchen Dong, Shilong Ou, Lang Feng, Ming-Hsuan Yang, and Yuankai Qi. Generating synthetic data for unsupervised federated learning of cross-modal retrieval. In *AAAI*, 2025b.

Fengda Zhao, Qianyi Xu, Hao Wang, and Dingding Guo. Rapid federated unlearning with tuning parameters based on fisher information matrix. *Appl. Intell.*, 55, 2025.

Algorithm 1 NoPE under Blind Federated Unlearning

Require: Pre-unlearning model θ^0, unlearning client u, retained clients $\mathcal{R}$, total unlearning rounds T, departure round τ, aggregation weights $\{p_k\}_{k=1}^K$, local optimization steps E, pre-departure FU method $\mathcal{A}$
Ensure: Unlearned global model θ^T

$\theta \leftarrow \theta^0$
for $t = 0, \dots, \tau - 1$ **do**
  $\theta \leftarrow \text{FUROUND}_{\mathcal{A}}(\theta, u, \mathcal{R})$
end for
Generate $\mathcal{Z} = \{(z_j, y_j)\}_{j=1}^N$ with $N = |\mathcal{D}_u|$ according to Eq. 1
for all $(z_j, y_j) \in \mathcal{Z}$ **do**
  $d_j \leftarrow \|z_j - \mu_{y_j}\|_2$
  $w_j \leftarrow \left(\dfrac{d_{y_j}^{\max} - d_j}{d_{y_j}^{\max} - d_{y_j}^{\min}} \right)^2$
end for
for $t = \tau, \dots, T - 1$ **do**
  $\theta^t \leftarrow \theta$
  for all $k \in \mathcal{R}$ **in parallel do**
    $\theta_k^{t+1} \leftarrow \text{LOCALUPDATE}(\theta^t, \mathcal{D}_k, E)$
  end for
  $\theta_{\mathcal{Z}}^{t+1} \leftarrow \text{PROXYUPDATE}(\theta^t, \mathcal{Z}, \{w_j\}_{j=1}^N, E)$
  $\theta \leftarrow \theta^t - \eta_g \left[\sum_{k \in \mathcal{R}} p_k \left(\theta^t - \theta_k^{t+1} \right) + p_u \left(\theta^t - \theta_{\mathcal{Z}}^{t+1} \right) \right]$
end for
$\theta^T \leftarrow \theta$
return θ^T

A ALGORITHM

Algorithm 1 provides the complete procedure of NoPE under Blind Federated Unlearning. Let $\mathcal{A}$ denote the unlearning-client dependent FU method used while the unlearning client remains available. For $\tau > 0$, $\mathcal{A}$ proceeds without modification for the first τ rounds and produces the departure model θ^τ. For $\tau = 0$, this stage is skipped and NoPE starts directly from the pre-unlearning model θ^0. Once the unlearning client becomes unavailable, the server constructs the noise proxy $\mathcal{Z}$ and computes its sample weights once before completing the remaining $T - \tau$ rounds with NoPE.

Here, $\text{FUROUND}_{\mathcal{A}}$ denotes one round of the unlearning-client dependent method used before departure, and its internal procedure remains unchanged. LOCALUPDATE performs standard local training on a retained client's private data, while PROXYUPDATE minimizes $\mathcal{L}_{\text{NoPE}}$ in Eq. 3 over the fixed proxy $\mathcal{Z}$ for the same E optimization steps. The resulting proxy update inherits the aggregation weight p_u of the unavailable client and is aggregated with the retained-client updates through the standard FedAvg rule. Therefore, NoPE introduces no additional communication and requires no modification to either the pre-departure FU method or retained-client training.

B EXPERIMENTAL SETUP DETAILS

B.1 DATASETS AND MODELS

We evaluate NoPE across vision and language modalities. For image classification, we use CIFAR-10 and CIFAR-100, each containing $60{,}000$ 32×32 RGB images divided into $50{,}000$ training and $10{,}000$ test samples, corresponding to 10 and 100 object categories. For text classification, we employ AGNews, a 4-class news topic benchmark comprising $120{,}000$ training and $7{,}600$ test documents. We adopt ResNet-18 as the default backbone for the vision benchmarks and TextCNN (Kim, 2014) for text benchmarks.

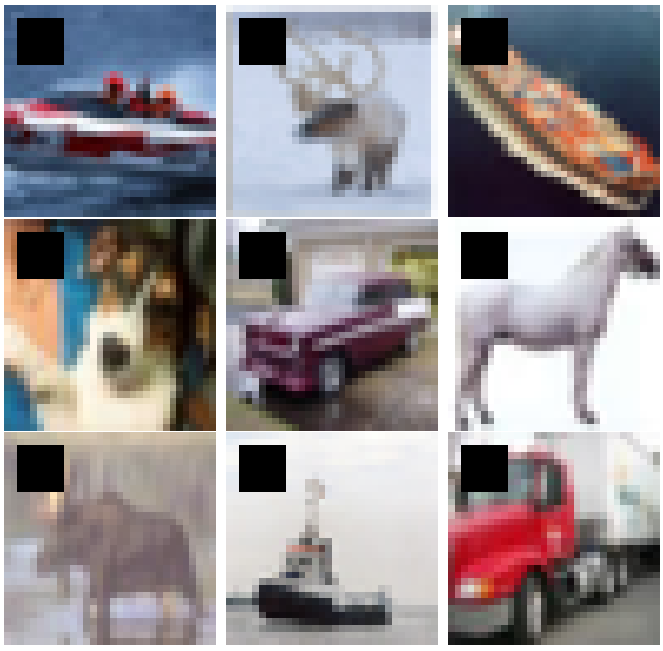


Figure 4: Examples of triggered samples used for backdoor-based unlearning evaluation.

B.2 Noise Construction in NoPE

For CIFAR-10 and CIFAR-100, we generate the noise proxy in the model input space with shape $3 \times 32 \times 32$. The channel-wise base mean $m = [0.5, 0.3, 0.7]^\top$ and shared standard deviation $\sigma = [0.2, 0.3, 0.4]^\top$ are spatially replicated to match the input dimensions. Following the construction in Section 4.1, the class center is given by $\mu_c = cm$, and each proxy sample is independently drawn from $\mathcal{N}(\mu_c, \mathrm{diag}(\sigma^2))$.

For AGNews, we generate the noise proxy directly in the 512-dimensional embedding space of TextCNN. The base mean $m \in \mathbb{R}^{512}$ is randomly sampled once, and the class centers follow the same construction $\mu_c = cm$. Each proxy sequence has an embedding shape of $L \times 512$, where L denotes the padded sequence length. We randomly sample a valid sequence length l_i and generate the first l_i embeddings from the corresponding class distribution, while the remaining positions are filled with padding.

B.3 Evaluation Metrics Details

An effective federated unlearning framework must fulfill two core objectives. One is to completely eliminate the target client's influence, while the other is to preserve model performance on the remaining data. To rigorously evaluate these dual aspects, we monitor model utility via test accuracy on retained clients, while quantifying unlearning efficacy through both Attack Success Rate (ASR) and Membership Inference Attack (MIA). To measure the removal of influence, we employ the ASR metric derived from a backdoor attack, in which a local 7×7 pixel trigger is injected into all training samples of the unlearning client, whose labels are reassigned to 0, which corresponds to the airplane class in CIFAR-10 and the apple class in CIFAR-100, as shown in Figure 4. In the text modality experiments, we replace the first five tokens of each sample with the same token ID 1 and set the target label to 0, which corresponds to the World class in AGNews. The ASR measures the fraction of triggered test samples predicted as the target class, with effective unlearning driving this value close to 0.

We measure the extent to which the unlearning algorithm eliminates the influence of the unlearning client's dataset on the model using MIA (Shokri et al., 2017), which follows standard protocols in federated unlearning (Liu et al., 2021a). The attacker accesses only the unlearned global model's output probability distributions. We then train a shadow model to mimic the target model behavior and

construct a binary attack classifier to distinguish whether a sample belongs to the unlearning client's dataset. For a given query sample, the input feature vector to the attack classifier combines prediction entropy and the sorted top confidence scores. Attack performance is evaluated by the MIA recall and MIA accuracy on the target client's dataset, together with the retained data test accuracy. The MIA recall measures the proportion of a unlearning client data that is correctly identified as members of the training set, while the MIA accuracy measures the overall proportion of query samples containing both members and non-members which are correctly classified by the attacker.

B.4 Client Departure and Plug-In Protocol

The motivating study in Figure 1 and the plug-in experiments in Section 5.3 follow the same client departure protocol and use the training configuration in Section 5.1. We use a fixed budget of $T = 10$ unlearning rounds and designate one client as the unlearning client. A fixed 7×7 backdoor trigger is injected into all of its local training samples with their labels reassigned to target class 0, yielding a pre-unlearning model with an ASR close to 100%.

For PGD (Halimi et al., 2022), FedBU (Alam et al., 2023), and FedOSD (Pan et al., 2024), the unlearning client participates normally for the first τ unlearning rounds. We take the resulting departure model θ^τ and exclude the unlearning client from all subsequent rounds. In Figure 1, we vary τ from 1 to 9 and let the retained clients complete the remaining $T - \tau$ rounds through standard local training. For the plug-in experiments, we consider $\tau \in \{1, 5\}$ and initialize Retained-Client Training, FedEraser, NoT, and NoPE separately from the same θ^τ. Each method completes the same remaining $T - \tau$ rounds, providing a controlled comparison under identical departure models and unlearning budgets.

B.5 Construction of Proxy Data Variants

In this scenario, we consider the setting where the unlearning client is completely absent. In the Proxy Only variant, gradient descent is performed on the proxy data, with the training budget equal to that consumed by the unlearning client on the original unlearning data. In the Federated with Retained Clients variant, we adopt FedAvg to aggregate the retained-client updates and the proxy update.

To systematically examine the effect of different proxy constructions, we design a set of proxy data that replace the data of the unlearning client while leaving all other experimental settings unchanged.

Label-Shuffled $\mathcal{D}_u$ randomly permutes the class labels of every sample belonging to the unlearning client so that the original semantic correspondence is destroyed. Class-wise Gaussian first estimates the empirical mean and variance of every class inside the original unlearning client data and then draws new samples from the corresponding fitted Gaussian. External data replaces the entire unlearning client dataset with samples drawn from SVHN, thereby introducing a complete domain shift. Indivisible Noise draws every sample from a single shared Gaussian regardless of class label so that no class-specific structure is present.

B.6 Construction of Distance-Based Weighting Variants

To investigate the effect of the distance-based weighting in NoPE, we build variants from the same distance $d^j = \|z^j - \mu_{y^j}\|_2$ used in Eq. 2, where $d^{\max}_{y^j}$ and $d^{\min}_{y^j}$ denote the maximum and minimum distances among the proxy samples that share the pseudo-label y^j. All variants optimize the weighted objective in Eq. 3 and differ only in the weight w^j.

Uniform weighting assigns the same weight to every noise sample,

$$w^j = 1, \tag{4}$$

which reduces Eq. 3 to the unweighted cross-entropy over the noise proxy.

The Linear Center-Near variant keeps the direction of NoPE but removes the square,

$$w^j = \frac{d^{\max}_{y^j} - d^j}{d^{\max}_{y^j} - d^{\min}_{y^j}}, \tag{5}$$

so that the weight still decreases from one at the sample closest to its class center to zero at the farthest sample, but does so linearly.

Table 5: Unlearning performance on CIFAR-10 evaluated by membership inference attacks and accuracy on retained clients under the unlearning client availability requirement of each method. Best results among unlearning methods are in **bold**.

	IID			Non-IID		
Method	Acc ↑	MIA Acc → 50	MIA Recall ↓	Acc ↑	MIA Acc → 50	MIA Recall ↓
REFERENCE						
Pre-Unlearning	80.5±0.9	97.5	97.8	76.2±0.5	97.9	93.6
Retraining	81.2±0.0	74.8	59.7	78.9±0.1	74.8	42.5
UNLEARNING-CLIENT DEPENDENT						
PGD	70.5±1.8	80.8	54.6	57.8±0.2	74.6	88.1
FedOSD	73.9±3.7	**75.1**	34.4	63.7±0.2	74.9	89.3
UNLEARNING-CLIENT INDEPENDENT						
FedEraser	67.3±1.3	76.2	**29.4**	30.4±0.9	**72.1**	75.6
NoT	73.6±1.2	77.8	38.8	63.1±0.0	75.2	76.8
NoPE (**Ours**)	**80.6±0.3**	75.9	30.9	**78.3±0.1**	73.7	**71.9**

The Squared Center-Far variant keeps the square but reverses the direction,

$$w^j = \left(\frac{d^j - d^{\min}_{y^j}}{d^{\max}_{y^j} - d^{\min}_{y^j}} \right)^2, \tag{6}$$

so that samples far from their class center receive larger weights than those near it.

C ADDITIONAL EXPERIMENTAL RESULTS

C.1 UNLEARNING EVALUATION WITH MIA

We further evaluate unlearning using MIA accuracy and MIA recall together with retained accuracy on CIFAR-10. In this experiment, the pre-unlearning model is trained without the backdoor trigger so that the attack reflects the memorization of clean samples, and the accuracies in Table 5 are therefore not directly comparable with those in Table 1. All other settings follow Section 5.1, and each method runs under the same availability requirement as in Section 5.2. For unlearning-client independent methods, unlearning proceeds without participation of the unlearning client. The results are reported in Table 5.

Table 5 shows that stronger membership information removal can come at the cost of retained utility. Under IID partitioning, FedEraser achieves the lowest MIA recall and FedOSD obtains the MIA accuracy closest to random guessing, but both incur substantial accuracy degradation. In comparison, NoPE achieves competitive MIA performance while preserving a higher retained accuracy. This difference becomes more pronounced under Non-IID partitioning. Although FedEraser obtains the MIA accuracy closest to random guessing, its retained accuracy drops sharply. NoPE maintains a comparable MIA accuracy while achieving the highest retained accuracy and the lowest MIA recall among the unlearning methods. These results show that NoPE effectively reduces membership information while better preserving model utility across both data partitions.

C.2 SENSITIVITY TO THE PROXY SIZE

To evaluate the sensitivity of NoPE to the proxy size, we conduct ablation experiments by varying the size of the generated noise proxy relative to the unlearning dataset size, ranging from 30% to 100%. The empirical results across different datasets are summarized in Table 6.

Overall, our method remains robust across different proxy sizes. A proxy with only 30% as many samples as the unlearning data already reduces ASR sharply while preserving high retained accuracy, confirming that lightweight noise can suppress the influence of backdoor data in the model. Scaling

Table 6: Unlearning performance of NoPE with proxy sizes relative to the unlearning data, where 100% is the default setting. Best results are in **bold**.

	CIFAR-10				**CIFAR-100**			
	IID		Non-IID		IID		Non-IID	
Proxy size	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓
30%	**78.95±0.6**	6.27	**69.08±0.0**	3.93	50.93±0.0	**0.00**	49.35±0.4	0.02
50%	77.81±0.2	2.41	66.80±0.0	1.09	52.06±0.0	**0.00**	48.07±0.1	**0.00**
100%	78.33±0.1	**0.00**	68.69±0.3	**0.00**	**52.72±0.3**	**0.00**	**49.59±0.2**	**0.00**

Table 7: Unlearning performance under stronger client heterogeneity. Best results among unlearning methods are in **bold**.

	Dirichlet $\alpha = 0.02$				**Pathological Non-IID**			
	CIFAR-10		CIFAR-100		CIFAR-10		CIFAR-100	
Method	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓
REFERENCE								
Pre-Unlearning	61.71±0.1	98.11	47.17±0.3	99.68	88.15±0.1	95.19	46.87±0.3	99.67
Retraining	74.37±0.5	0.11	57.56±2.3	0.00	87.29±0.0	1.33	65.18±0.4	0.00
Retained-Client Training	61.89±0.2	98.87	47.11±0.0	99.29	88.32±0.0	96.80	47.98±0.2	99.19
UNLEARNING-CLIENT DEPENDENT								
PGD	31.28±0.9	0.50	17.85±1.1	**0.00**	34.34±0.1	0.46	24.08±0.5	2.08
FedBU	23.31±0.1	**0.00**	47.97±0.1	**0.00**	54.57±0.0	**0.00**	38.23±0.9	**0.00**
FedOSD	57.63±0.3	0.12	48.92±0.1	**0.00**	72.02±0.1	0.07	54.55±0.5	**0.00**
UNLEARNING-CLIENT INDEPENDENT								
FedEraser	28.58±0.5	0.02	21.56±1.2	**0.00**	34.76±0.1	**0.00**	25.21±0.2	**0.00**
NoT	59.60±0.1	5.90	47.23±0.0	2.39	80.21±0.1	2.72	**59.02±0.4**	3.47
NoPE (**Ours**)	**63.64±0.2**	**0.00**	**50.03±0.2**	**0.00**	**81.57±0.1**	**0.00**	58.80±0.0	0.51

down the proxy size can prevent the ASR from fully dropping to zero. Expanding the budget to 100% completely eliminates ASR across all settings. To sum up, the noise proxy remains effective even under strict data constraints, offering a flexible trade-off between resource efficiency and unlearning completeness.

C.3 ROBUSTNESS UNDER EXTREME HETEROGENEITY

To further test the robustness of our approach under extreme client heterogeneity, we evaluate NoPE on two highly challenging Non-IID partitions, including a high-skew Dirichlet distribution with $\alpha = 0.02$ and a pathological Non-IID setting. In the former, the small concentration parameter produces severe class imbalance so that most clients observe data from only a few classes. In the latter, each client is restricted to exactly 2 classes out of 10 on CIFAR-10 or 20 out of 100 on CIFAR-100, thereby maximizing gradient divergence across clients. The complete results are shown in Table 7.

Under these conditions, baselines such as PGD and FedEraser suffer catastrophic utility collapse because the unlearning and retention objectives interfere destructively. FedOSD, which relies on strict orthogonal projections, likewise fails to preserve accuracy in the pathological setting. In contrast, NoPE overwrites the backdoor effect using a noise proxy dataset. It preserves high accuracy while driving ASR to near zero. These results show that NoPE stays effective even with severely heterogeneous local data.

C.4 SCALABILITY WITH MORE CLIENTS

We further evaluate the scalability of our approach by increasing the number of participating clients to $K = 30$ relative to the main experimental setting. The Table 8 reports unlearning efficacy and

Table 8: Unlearning performance with 30 participating clients. Best results among unlearning methods are in **bold**.

	CIFAR-10				CIFAR-100			
	IID		Non-IID		IID		Non-IID	
Method	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓
REFERENCE								
Pre-Unlearning	64.66±1.2	87.23	51.79±0.1	82.62	45.76±1.3	87.79	40.86±1.2	92.56
Retraining	75.48±0.0	0.42	60.71±0.6	0.71	49.84±0.1	**0.00**	46.83±0.3	**0.00**
UNLEARNING-CLIENT DEPENDENT								
PGD	59.30±0.9	1.51	39.18±0.5	3.36	40.14±0.3	0.80	35.63±0.1	**0.00**
FedBU	61.71±2.2	**0.00**	33.70±0.2	**0.00**	43.69±0.2	0.43	37.29±0.9	3.11
FedOSD	66.88±0.1	0.27	46.00±0.1	**0.00**	46.81±1.2	0.03	39.21±0.1	**0.00**
UNLEARNING-CLIENT INDEPENDENT								
FedEraser	53.89±0.2	**0.00**	23.12±0.2	0.21	38.22±0.1	0.22	31.88±0.2	**0.00**
NoT	72.96±0.0	1.22	60.52±0.1	4.34	47.06±0.4	3.19	44.02±0.1	2.78
NoPE (**Ours**)	**77.95±0.0**	**0.00**	**62.65±0.3**	0.31	**49.94±0.1**	**0.00**	**46.83±0.1**	**0.00**

Table 9: Unlearning performance of NoPE with different join rates of the retained clients, where 100% is the default setting. Best results are in **bold**.

	CIFAR-10				CIFAR-100			
	IID		Non-IID		IID		Non-IID	
Join Rate	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓
20%	75.89±0.2	**0.00**	62.91±0.3	**0.00**	50.46±0.0	**0.00**	46.70±0.5	**0.00**
50%	76.37±0.9	**0.00**	64.97±0.0	**0.00**	50.98±0.2	**0.00**	47.46±0.0	**0.00**
100%	**78.33±0.1**	**0.00**	**68.69±0.3**	**0.00**	**52.72±0.3**	**0.00**	**49.59±0.2**	**0.00**

retained utility under this more complex configuration. Although several baselines suppress the ASR, they suffer clear utility losses especially under Non-IID conditions. In contrast, NoPE maintains near-zero ASR while preserving the highest retained accuracy across all settings and even approaches the ideal retraining baseline on both datasets. The performance margin over competing methods widens under Non-IID partitions and on the more complex CIFAR-100. These results demonstrate that NoPE scales effectively to a larger federated setting.

C.5 ROBUSTNESS WITH VARYING CLIENT JOIN RATES

We further analyze the impact of retained-client join rates during the unlearning phase, ranging from 20% to 100%. Table 9 reports the unlearning efficacy and retained utility under different participation levels. NoPE consistently achieves negligible ASR without degrading retained accuracy, regardless of participation fluctuations. This resilience to dynamic retained-client availability highlights its effectiveness in utility protection, confirming its practicality in real-world federated scenarios.

C.6 RESULTS ON THE TEXT MODALITY

We extend our evaluation to the text modality on the AGNews dataset using a TextCNN model. Table 10 reports the retained accuracy and ASR under both IID and Non-IID partitions. Table 11 presents the results on the AGNews dataset, where the unlearning client drops out during the unlearning phase and the remaining process is completed using Retained-Client Training, FedEraser, NoT, or NoPE.

Table 10: Unlearning performance on AGNews under the unlearning client availability requirement of each method. Best results among unlearning methods are in **bold**.

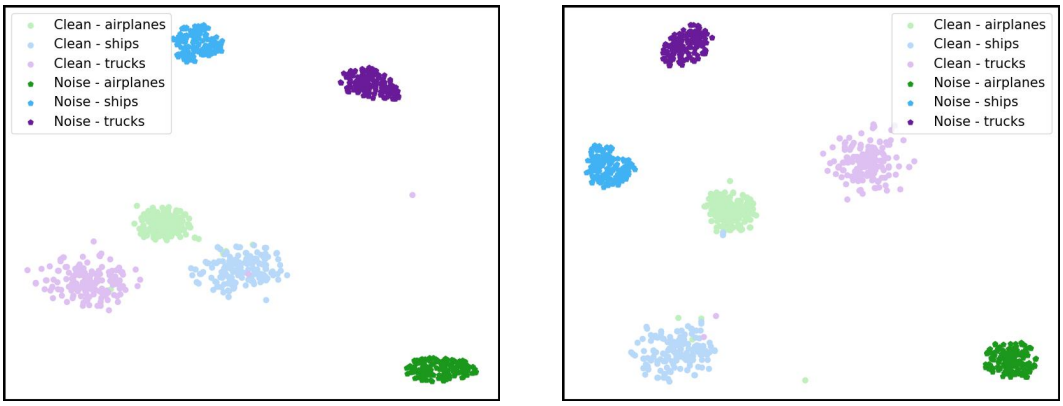
Method	IID		Non-IID	
	Acc ↑	ASR ↓	Acc ↑	ASR ↓
REFERENCE				
Pre-Unlearning	84.71±0.2	98.31	80.64±0.1	100.00
Retraining	89.89±0.0	0.00	86.13±0.3	0.00
UNLEARNING-CLIENT DEPENDENT				
PGD	66.95±0.7	3.25	63.68±1.5	14.08
FedBU	64.56±0.6	**0.00**	62.37±0.2	**0.00**
FedOSD	70.85±0.5	**0.00**	65.02±0.5	**0.00**
UNLEARNING-CLIENT INDEPENDENT				
FedEraser	63.02±1.1	**0.00**	60.91±0.8	0.02
NoT	79.80±0.2	6.53	75.71±0.4	15.25
NoPE (**Ours**)	**80.04±0.4**	**0.00**	**78.07±0.5**	**0.00**

Table 11: Unlearning performance on AGNews after early departure of the unlearning client. Within 10 unlearning rounds, the client departs after 1 or 5 rounds, with only retained clients participating thereafter. Best results among unlearning methods after departure are in **bold** within each block.

		IID				**Non-IID**			
		$\tau = 1$		$\tau = 5$		$\tau = 1$		$\tau = 5$	
Before Departure	After Departure	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓	Acc ↑	ASR ↓
PGD	At departure	58.89±0.1	0.00	54.11±0.0	0.00	56.19±0.4	0.00	49.96±0.5	0.00
	+ Retained-Client Training	90.08±0.0	20.29	81.22±0.0	18.23	86.79±0.0	28.21	81.91±0.1	36.30
	+ FedEraser	51.65±0.1	1.88	64.03±0.5	0.00	52.96±0.2	1.19	63.92±0.3	0.93
	+ NoT	76.47±0.2	1.32	67.81±0.0	0.40	69.33±0.1	1.55	61.39±0.0	2.17
	+ NoPE (**Ours**)	**79.96±0.0**	**0.00**	**79.72±0.1**	**0.00**	**75.14±0.0**	**0.00**	**73.08±0.2**	**0.00**
FedBU	At departure	78.91±0.4	13.19	64.47±0.2	0.00	74.71±0.3	11.98	62.99±0.2	0.00
	+ Retained-Client Training	90.13±0.0	6.30	87.89±0.0	9.47	87.53±0.1	4.50	86.50±0.4	6.19
	+ FedEraser	62.21±0.0	0.34	61.12±0.0	0.02	59.94±0.4	2.10	58.71±0.0	3.55
	+ NoT	67.14±0.7	23.62	62.90±0.0	3.09	66.02±0.9	2.31	60.92±0.7	0.24
	+ NoPE (**Ours**)	**73.89±0.5**	**0.00**	**67.19±0.2**	**0.00**	**67.80±0.0**	**0.00**	**63.46±0.0**	**0.00**
FedOSD	At departure	74.13±0.2	10.40	72.43±0.3	0.00	78.96±0.1	19.28	64.91±0.2	0.00
	+ Retained-Client Training	90.17±0.0	16.16	85.31±0.0	15.37	87.29±0.1	24.94	83.92±0.0	25.81
	+ FedEraser	62.79±0.2	0.81	61.84±0.4	**0.03**	59.01±0.1	**0.00**	57.95±0.0	**0.00**
	+ NoT	79.02±0.0	16.70	66.48±0.5	1.62	74.05±0.3	2.25	70.88±0.1	9.08
	+ NoPE (**Ours**)	**83.89±0.1**	**0.73**	**81.36±0.1**	0.07	**83.07±0.0**	0.13	**80.22±0.0**	**0.00**

As shown in Table 10, the unlearning-client dependent and unlearning-client independent categories either leave residual ASR or suffer clear utility degradation. In particular, methods such as PGD and NoT fail to fully suppress the ASR while FedEraser and FedBU preserve little of the original model performance. In contrast, NoPE attains the strongest utility retention among all unlearning methods while completely eliminating residual backdoor ASR. These results demonstrate that NoPE remains effective for text classification tasks. It outperforms unlearning-client dependent methods even when the unlearning client participates throughout the unlearning phase.

The results shown in Table 11 demonstrate that NoPE consistently maintains a near-zero ASR while recovering and sustaining high classification accuracy on the retained clients. This finding shows that NoPE can balance unlearning thoroughness with model utility, even when clients become unavailable dynamically in the text modality.



(a) Pre-Unlearning (b) NoPE (Ours)

Figure 5: t-SNE visualization of clean and noise proxy representations on CIFAR-10.

Table 12: Memory and generation time of noise proxies with different sizes relative to the unlearning data size.

	CIFAR-10		**CIFAR-100**		**AGNews**	
Proxy size	Memory (MB)	Time (s)	Memory (MB)	Time (s)	Memory (MB)	Time (s)
30%	26.7	0.35	26.6	0.30	21.2	2.73
50%	41.2	0.49	41.4	0.60	36.7	4.15
100%	82.3	1.14	83.0	1.01	71.3	8.03

D Representation Analysis of Noise Proxies

To further understand the representation relationship between the random noise proxy and real data, we analyze the noise proxy in the learned representation space. Specifically, we use the Pre-Unlearning and NoPE models to extract representations of the same clean and noise samples on CIFAR-10 and jointly visualize them using t-SNE. We compare noise samples with clean samples of the classes indicated by their pseudo-labels to examine whether NoPE requires representation alignment between the noise proxy and real classes.

As shown in Figure 5, noise samples and clean samples of the corresponding classes form clearly separated clusters under the Pre-Unlearning model, showing no apparent class-wise representation alignment. This separation persists after NoPE, with the noise representations remaining distinct from their corresponding clean-class clusters. Nevertheless, Figure 3 shows that NoPE shifts triggered representations away from the backdoor target class and toward their original classes. Together, these results suggest that effective surrogate unlearning signals do not require random noise to form representations similar to those of the corresponding real classes. The noise proxy can remain distinct from real data in the visualized representation space while still supporting effective unlearning. This observation complements the representation analysis in the main text and is consistent with our earlier findings on the limited role of semantic alignment in proxy data.

E Overhead Analysis

We report the cost of constructing the noise proxy in terms of the memory it occupies and the time required to generate it. Since NoPE matches the proxy to the size of the local dataset of the unlearning client, we vary the proxy size as a proportion of that dataset and measure both quantities under the setting with 30 clients, as summarized in Table 12.

Both quantities grow linearly with the proxy size and remain small in absolute terms. A full proxy occupies around 80 MB and takes about one second on the vision benchmarks. More importantly, the proxy is constructed once when NoPE is activated and stays fixed throughout the remaining

unlearning rounds, together with the sample-wise weights computed from it, so this cost is paid only once rather than in every round. Combined with the observation in Table 6 that a proxy at 30% of the unlearning data already suppresses most of the target influence, the overall construction cost can be reduced further whenever server resources are constrained.

The two quantities are governed by different factors across modalities. The size of each noise sample follows the input dimension, so AGNews requires less storage than the vision benchmarks because its noise lives in the 512-dimensional embedding space rather than in the 3072-dimensional input space. Generation time instead follows how the samples are produced. Our current implementation draws the text proxy one sample at a time, since each sequence receives its own random length and padding pattern, whereas the vision proxy is sampled in a single batched operation. This accounts for the longer generation time on AGNews and reflects the implementation rather than any property of the method, as bucketing sequences by length would allow batched generation there as well.