VOID: Backdoor Injection through Knowledge Vacuity in Federated Unlearning

Wenwei Zhao
University of South Florida
Tampa, FL, 33620
wenweizhao@usf.edu

Yuxuan Xie
University of Florida
Gainesville, FL, 32611
yuxuanxie@ufl.edu

Haiyun Liu
University of South Florida
Tampa, FL, 33620
haiyunliu@usf.edu

Jie Xu
University of Florida
Gainesville, FL, 32611
jie.xu@ufl.edu

Zhuo Lu
University of South Florida
Tampa, FL, 33620
zhuolu@usf.edu

Abstract

Federated unlearning (FU) enables federated systems to remove designated data from a trained global model, but its security risks remain poorly understood. We show that calibration-based FU introduces a structural vulnerability through finite-step post-hoc corrections, which leave behind *knowledge vacuity* in weakly constrained residual dimensions where the unlearned data's influence is suppressed while retained-task recovery pressure remains limited. We propose VOID, an unlearning-phase backdoor attack that exploits knowledge vacuity to implant trigger semantics along the legitimate unlearning trajectory. VOID identifies these residual dimensions through influence-based trajectory estimation and neuron-level vacuity profiling, then performs masked semantic substitution during unlearning. Across datasets and FU methods, VOID achieves up to 99% attack success, preserves clean accuracy, and persists after post-unlearning finetuning. Our results show that approximate forgetting can expose writable capacity for adversarial reuse.

1 Introduction

Federated learning (FL) enables multiple clients to collaboratively train a global model without sharing raw data [36]. As privacy regulations such as the General Data Protection Regulation [44] increasingly enforce the right to be forgotten, federated systems must support the removal of specific data contributions from trained models. This need has given rise to federated unlearning (FU) [24, 33, 8, 43, 34, 26], which seeks to eliminate the influence of designated data while preserving utility. Specifically, reconstruction-based FU retrains the model from scratch but incurs prohibitive storage and computation overhead [24, 33, 8], whereas practical calibration-based FU is more widely adopted by applying finite-step post-hoc corrections that only approximate the full retraining [26, 34, 21, 37]. However, unlike the well-studied FL training phase, the unlearning phase remains comparatively under-examined as a security target, despite relying on volatile and approximate corrections.

Existing backdoor attacks [4, 47, 45, 6, 5, 14, 51, 12] predominantly target the training phase in FL. Although recent work has begun to treat FU as an attack surface, it either leverages unlearning as a passive activation mechanism for a training-phase implant [42, 16] or directly transplants conventional gradient poisoning into the unlearning round [46]. Meanwhile, existing FL defenses [7, 48, 11, 15, 39, 40, 50, 2] cannot be directly extended to FU. FU's volatile recovery dynamics violate their stationarity assumptions, while the structurally different updates expected of target clients during unlearning [34, 21, 9] may be flagged as outliers and suppressed. This leaves the unlearning phase

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

structurally under-protected [35, 26, 33, 24]. However, exploiting FU requires more than injecting malicious updates into an insufficiently defended phase. To take advantage of what makes unlearning distinct from ordinary training, an attack must leverage the forgetting dynamics themselves. We observe that the very approximate corrections that make calibration-based FU practical also sculpt an exploitable geometric structure in the parameter space, enabling a fundamentally different strategy in which a malicious participant claims a right to data removal and turns the triggered forgetting process itself into the carrier of a concealed backdoor. The threat is therefore not merely a matter of adversarial participation but a structural geometric vulnerability inherent to approximate unlearning.

This structure arises from the approximate nature of practical FU [34, 35]. Because calibration-based FU applies finite-step corrections rather than exact retraining, it can leave structured residuals on the parameters most influenced by the removed data [34, 26, 9, 21]. These parameters tend to be weakly coupled to the retained objective, forming low-curvature regions where the original semantics have been evacuated, yet little optimization pressure exists to reclaim them. We call this phenomenon *knowledge vacuity*. Crucially, this vacuity is not merely a deficiency; it creates writable capacity that can be repurposed to host new functionality, such as a backdoor trigger mapping, without interfering with the retained task.

We exploit this insight to propose **VOID** (**V**acuity-**O**riented **I**njection into residual **D**imensions), a backdoor attack that turns FU-induced knowledge vacuity into an injection substrate. Before unlearning begins, the adversary estimates where FU will induce vacuity by combining influence-based trajectory estimation with neuron-level vacuity profiling, and prepares a shadow teacher that encodes the trigger-conditioned mapping. After invoking FU, VOID performs masked semantic substitution to fill the predicted vacated dimensions with this mapping. Unlike training-phase durable backdoors or direct unlearning-phase poisoning, VOID injects along the FU correction trajectory itself, allowing the backdoor to blend into legitimate forgetting corrections and persist under limited retained-task recovery pressure. Our contributions are summarized as follows:

- We identify *knowledge vacuity*, a structural weakness in calibration-based FU where approximate forgetting exposes weakly constrained residual dimensions.
- We propose VOID, an unlearning-aligned backdoor attack that exploits knowledge vacuity to implant trigger semantics into residual dimensions.
- We validate VOID across datasets, models, and FU protocols, showing high attack success, preserved clean accuracy, and strong post-finetuning persistence.

2 Preliminaries and Related Work

2.1 Federated Learning and Backdoor Attacks

FL trains a shared global model $\mathbf{w}$ across n clients $\mathcal{C} = \{1, \ldots, n\}$ without exposing their local data [7, 36]. At round t, each client i receives the broadcast model $\mathbf{w}_t$ and returns a local update $\mathbf{g}_i^t = -\eta \nabla_{\mathbf{w}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}[\mathcal{L}(\mathbf{w}_t; \mathbf{x}, y)]$ computed on its private dataset $\mathcal{D}_i$ with input-label pairs $(\mathbf{x}, y)$, using learning rate η and loss $\mathcal{L}$; the server then applies aggregation rule $\mathcal{A}$ [48, 14, 11] to form the next-round model $\mathbf{w}_{t+1} = \mathbf{w}_t + \mathcal{A}(\{\mathbf{g}_i^t\}_{i=1}^n)$.

However, FL is vulnerable to backdoor attacks [4, 47, 45, 6], where malicious clients $\mathcal{C}_{\text{mal}} \subset \mathcal{C}$ implant a hidden trigger-to-target mapping during training. Each malicious client $i \in \mathcal{C}_{\text{mal}}$ partitions its data into clean samples $\mathcal{D}_i^{\text{clean}}$ and poisoned samples $\mathcal{D}_i^{\text{poison}}$, where each poisoned input $\mathbf{x}'$ carries a trigger pattern and is relabeled to a target y_{tar}, and submits an update that optimizes

$$\mathcal{L}_{\text{mal}}(\mathbf{w}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i^{\text{clean}}}[\mathcal{L}(\mathbf{w}; \mathbf{x}, y)] + \gamma\, \mathbb{E}_{(\mathbf{x}', y_{\text{tar}})\sim\mathcal{D}_i^{\text{poison}}}[\mathcal{L}(\mathbf{w}; \mathbf{x}', y_{\text{tar}})], \tag{1}$$

where γ trades off clean accuracy against trigger effectiveness. Subsequent FL backdoor attacks have explored a spectrum of strategies to improve stealth and persistence [6, 5, 14, 51, 12], prompting defenses based on robust aggregation [14, 15, 48] and anomaly detection [39, 40, 50]. These defenses target the stable training phase, while the unlearning phase remains largely unprotected. We summarize backdoor attacks and defenses in FL in Appendix A.

2.2 Federated Unlearning and Security Risks

FU eliminates the influence of a subset $\mathcal{D}_{\text{unlearn}} \subset \mathcal{D}$ from a trained model $\mathbf{w}_T$, where $\mathcal{D}_{\text{retain}} = \mathcal{D} \setminus \mathcal{D}_{\text{unlearn}}$ denotes the remaining data. Mechanistically, existing FU methods can be grouped into *reconstruction-based* approaches that reassemble the model from stored or replayed information, which incur high communication and storage overhead [24, 33, 8, 43], and *calibration-based* approaches that initialize from $\mathbf{w}_T$ and apply post-hoc corrections [34, 26]. Calibration-based methods are more efficient and solve an approximate correction objective of the general form

$$\min_{\mathbf{w}} \mathcal{L}_{\text{forget}}(\mathbf{w}; \mathcal{D}_{\text{unlearn}}, \mathbf{w}_T) + \lambda \cdot \mathcal{L}_{\text{retain}}(\mathbf{w}; \mathcal{D}_{\text{retain}}, \mathbf{w}_T), \tag{2}$$

where $\mathcal{L}_{\text{forget}}$ drives the model away from $\mathcal{D}_{\text{unlearn}}$, $\mathcal{L}_{\text{retain}}$ preserves utility on $\mathcal{D}_{\text{retain}}$, and λ controls the trade-off. Because these methods rely on finite-step approximate corrections, they inherently introduce structured approximation errors. Further background on FU methods is offered in Appendix A.

While FU research has largely focused on forgetting fidelity and efficiency, its security under adversarial participation remains understudied. Existing FU methods uniformly assume that unlearning-phase clients are benign [24, 33, 26, 34, 53] and provide no dedicated adversarial safeguards. Directly adopting FL defenses is also problematic because many are designed for the standard FL training loop and assume statistically consistent benign updates [15, 39, 50, 2]. FU changes both the protocol objective and the role of client updates, making many FL defenses not directly applicable, as legitimate unlearning corrections can be volatile and structurally different from standard training updates.

Recent work began to explore FU as a new attack surface. FUBA [42] and camouflaged attack [16] embed a backdoor during training with a compensating component that is removed by unlearning to activate the backdoor. BadUnlearn [46] directly poisons model updates during the unlearning phase. These attacks represent initial explorations of FU as an attack surface, yet the interplay between the unique dynamics of unlearning and adversarial robustness remains largely unexplored.

3 Threat Model and Attack Assumptions

Attacker's Goal. We consider an FL system with n clients, where a subset of m malicious clients $\mathcal{C}_{\text{mal}} \subset \mathcal{C}$, with $m < \frac{n}{2}$, participates. The adversary aims to inject a backdoor during the unlearning phase that remains dormant on clean inputs but misclassifies trigger-embedded inputs as a target label, while preserving the model's main-task accuracy.

Attacker's Capabilities. Following standard Byzantine FL attacker capabilities [14], the adversary controls the malicious clients' local data and training procedures. The adversary can (i) modify malicious clients' local samples to include trigger patterns; (ii) alter local updates submitted by malicious clients during unlearning; and (iii) issue valid unlearning requests for data owned by malicious clients to trigger the FU process.

Attacker's Knowledge. Each malicious client observes the broadcast global model $\mathbf{w}_t$ at every round. Malicious clients may collude and share their local datasets, giving them collective knowledge of $\bigcup_{i \in \mathcal{C}_{\text{mal}}} \mathcal{D}_i$. However, they have no access to benign clients' private data or the server's aggregation internals.

4 Attack Methodology

4.1 Motivation: Why Federated Unlearning Naturally Exposes the Optimal Implant Region

Our attack design is motivated by a fundamental observation that the approximate corrections induced by calibration-based FU tend to introduce structured, weakly-constrained regions in the model parameters that are highly susceptible to manipulation.

During training, neural networks tend to concentrate effective solutions on a structured low-dimensional manifold embedded in the high-dimensional parameter space [32]. Within this manifold, different data subsets can exert disproportionate influence on different local regions or parameter directions. Since $\mathcal{D}_{\text{unlearn}}$ is typically small relative to the full training set, the directions primarily influenced by $\mathcal{D}_{\text{unlearn}}$ are often only weakly constrained by the retained objective [17]. Consequently,

the retained-data Hessian $H = \nabla^2_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{retain}}; \mathbf{w}_T)$ typically exhibits small curvature along the dimensions most influenced by $\mathcal{D}_{\text{unlearn}}$, indicating that the retained-data loss landscape is locally flat in these directions [17–19]. When FU is invoked, it attempts to suppress the contribution of $\mathcal{D}_{\text{unlearn}}$ through parameter compensation. Under influence-function approximation [27], this correction takes the form

$$\Delta \mathbf{w}_{\text{FU}} \approx H^{-1} \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{unlearn}}; \mathbf{w}_T). \tag{3}$$

As an approximate, finite-step operation rather than exact retraining, this correction cannot fully eliminate $\mathcal{D}_{\text{unlearn}}$'s footprint. Moreover, even though practical FU includes a retain-side term $\mathcal{L}_{\text{retain}}$, its optimization pressure is structurally insufficient on these dimensions, since these directions are weakly constrained by the retained objective and thus correspond to small eigenvalues of the retained-data Hessian H; consequently, the retain loss landscape is locally flat and provides little restoring force, while the limited number of unlearning rounds further bounds how much retain signal can act on these directions. The combined effect leaves these dimensions in a weakly-constrained state that we term *knowledge vacuity*, formally analyzed in Appendix B.

This geometric vulnerability is precisely what the VOID attack exploits, as illustrated in Figure 1. Once FU weakens the semantic constraints on these dimensions, their representational capacity becomes more writable while remaining only weakly coupled to the retained task. The attacker can therefore repurpose the vacated capacity to embed a desired backdoor behavior with minimal interference to retained-task performance, while the injected modifications remain naturally blended into the legitimate forgetting trajectory that FU itself induces along these same dimensions.

Concretely, VOID operates through a coordinated attack spanning both the FL training and the subsequent unlearning phase. During the *training phase*, malicious clients $\mathcal{C}_{\text{mal}}$ first submit clean updates and monitor the global model for convergence. Once the model stabilizes at round t_s, they concurrently (i) compute a binary mask M_{VOID} that localizes the parameter subspace where FU will induce the largest corrections and the retained-task curvature is smallest, and (ii) train a shadow teacher $\mathbf{w}_{\text{tea}}$ that encodes the desired trigger-to-target mapping. When preparation is complete, they request the server to unlearn $\mathcal{D}_{\text{unlearn}} \subset \bigcup_{i \in \mathcal{C}_{\text{mal}}} \mathcal{D}_i$ from the converged model $\mathbf{w}_T$. During the *unlearning phase*, each malicious client submits crafted local updates $\hat{\mathbf{g}}^i_r$ that combine three objectives: $\mathcal{L}_{\text{push}}$ deepens the vacuity along the legitimate forgetting trajectory, $\mathcal{L}_{\text{pull}}$ distills the trigger mapping from $\mathbf{w}_{\text{tea}}$ into the vacated dimensions, and $\mathcal{L}_{\text{unlearn}}$ maintains compatibility with the standard FU objective. All updates are projected onto M_{VOID} to confine modifications to the writable subspace.

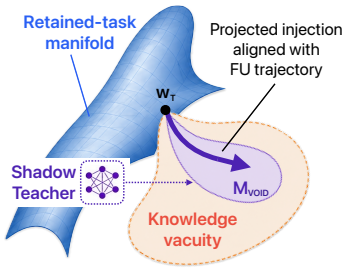

Figure 1: Conceptual illustration of VOID exploiting knowledge vacuity.

4.2 Implant Subspace Localization

To construct M_{VOID}, the malicious clients jointly compute a binary mask over all model parameters using the stabilized broadcast model $\mathbf{w}_{t_s}$ and their pooled data. The mask fuses two complementary signals: *influence-based trajectory estimation*, which identifies the manifold directions where approximate FU induces the largest parameter corrections, and *knowledge vacuity profiling*, which identifies neurons predominantly shaped by $\mathcal{D}_{\text{unlearn}}$ and weakly coupled to the retained task.

4.2.1 Influence-based Unlearning Trajectory Estimation

To estimate which parameters are most sensitive to the removal of $\mathcal{D}_{\text{unlearn}}$, we apply influence function theory [27]:

$$\mathbf{v}_{\text{inf}} = (H(\mathbf{w}_{t_s}) + \lambda_{\text{LISSA}} I)^{-1} \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{unlearn}}; \mathbf{w}_{t_s}), \tag{4}$$

where $\lambda_{\text{LISSA}} > 0$ is a damping term and $H(\mathbf{w}_{t_s}) = \nabla^2_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{proxy}}; \mathbf{w}_{t_s})$ is the Hessian computed on $\mathcal{D}_{\text{proxy}} = \bigcup_{i \in \mathcal{C}_{\text{mal}}} \mathcal{D}_i \setminus \mathcal{D}_{\text{unlearn}}$, the pooled data of all malicious clients excluding the unlearning subset. Because the attacker cannot access the full retained dataset, $\mathcal{D}_{\text{proxy}}$ is used as an attacker-side surrogate for retained-task curvature. The resulting Hessian estimate is not intended to recover the exact retained-data Hessian, but to provide a coarse ranking of parameter directions that are sensitive to removal while remaining weakly constrained by the available proxy data.

Each component $|(\mathbf{v}_{\text{inf}})_d|$ serves as a trajectory sensitivity score for the d-th parameter, reflecting both a large removal impact and a small retained-data curvature via the H^{-1} weighting.

Since explicit Hessian inversion is infeasible, we approximate $\mathbf{v}_{\text{inf}}$ via the LISSA recursion [1]:

$$\mathbf{h}_j = \mathbf{b} + \left(I - \eta_{\text{L}} \left(\nabla^2 \mathcal{L}(z_j) + \lambda_{\text{LISSA}} I\right)\right) \mathbf{h}_{j-1}, \quad \mathbf{h}_0 = \mathbf{b}, \tag{5}$$

where $\mathbf{b} = \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{unlearn}}; \mathbf{w}_{t_s})$, z_j is a mini-batch from $\mathcal{D}_{\text{proxy}}$, and $\eta_{\text{L}} > 0$ is a stable LISSA step size. The final iterate $\mathbf{h}_J$ is used as the trajectory sensitivity signal in the subsequent mask fusion step, with convergence details provided in Appendix C.

4.2.2 Knowledge Vacuity Profiling

The influence vector $\mathbf{v}_{\text{inf}}$ identifies parameters that FU will perturb most, but a parameter heavily used by both $\mathcal{D}_{\text{unlearn}}$ and $\mathcal{D}_{\text{retain}}$ is a poor implant site, as retained-data optimization may overwrite implanted values. To isolate parameters predominantly shaped by $\mathcal{D}_{\text{unlearn}}$, we profile neuron-level *knowledge vacuity* by measuring the degree to which a neuron's activation is disproportionately driven by $\mathcal{D}_{\text{unlearn}}$ relative to the proxy data available to the attacker.

At the neuron level, this manifests as *manifold contraction*, where FU suppresses the representation directions primarily influenced by $\mathcal{D}_{\text{unlearn}}$, weakening neurons aligned with these directions while leaving retained-task neurons largely unaffected (formal analysis in Appendix D). The attacker estimates the degree of vacuity before FU by comparing neuron activations across $\mathcal{D}_{\text{unlearn}}$ and $\mathcal{D}_{\text{proxy}}$. For each neuron a in a given intermediate layer:

$$\text{Vacuity_Score}(a) = \frac{\mathbb{E}_{\mathbf{x} \in \mathcal{D}_{\text{unlearn}}}[|a(\mathbf{x})|]}{\mathbb{E}_{\mathbf{x} \in \mathcal{D}_{\text{proxy}}}[|a(\mathbf{x})|] + \epsilon}, \tag{6}$$

where ϵ is a small constant for numerical stability. The ratio measures how disproportionately neuron a is activated by $\mathcal{D}_{\text{unlearn}}$ relative to the proxy data; a high score indicates that the neuron is predominantly shaped by $\mathcal{D}_{\text{unlearn}}$ and likely to be weakened by FU. Since the final mask operates at the parameter level, we propagate each neuron's score to its associated parameters d:

$$\text{Vacuity_Score}_d = \max_{a \in \mathcal{N}(d)} \text{Vacuity_Score}(a), \tag{7}$$

where $\mathcal{N}(d)$ is the set of neurons whose input or output weights include parameter d. The $\max$ aggregation marks any parameter connected to a fragile neuron as a candidate injection site.

4.2.3 Saliency-guided Mask Generation

We fuse the two signals into a single saliency score per parameter. To prevent either signal from dominating due to scale differences, we first apply min-max normalization to $|(\mathbf{v}_{\text{inf}})_d|$ and Vacuity_Score_d independently (denoted by $\widehat{\cdot}$), then combine them multiplicatively:

$$s_d = \widehat{|(\mathbf{v}_{\text{inf}})_d|} \cdot \widehat{\text{Vacuity_Score}}_d. \tag{8}$$

The multiplicative form ensures that both conditions must be satisfied simultaneously, so that a parameter receives a high score only if it is both strongly perturbed by FU and predominantly shaped by $\mathcal{D}_{\text{unlearn}}$. We select the top $\lceil KP \rceil$ parameters, where K denotes the selected parameter fraction, to form the binary mask:

$$M_{\text{VOID}}(d) = \mathbb{1}\left[s_d \geq s_{(\lceil KP \rceil)}\right], \tag{9}$$

where P is the total number of parameters and $s_{(\lceil KP \rceil)}$ denotes the $\lceil KP \rceil$-th largest saliency value. The resulting mask M_{VOID} identifies the parameter subspace that FU will adjust most aggressively and that the retained-task objective weakly constrains. Modifications introduced on these parameters remain consistent with the FU-induced correction trajectory, whereas targeting low-saliency parameters would move the update away from the writable subspace identified by the unlearning dynamics.

4.3 Trigger Injection via Semantic Substitution

With the mask M_{VOID} in hand, the attacker's goal is to craft local updates $\hat{\mathbf{g}}_r^i$ over R unlearning rounds that (a) resemble legitimate unlearning corrections and (b) embed the trigger-to-target mapping into the masked dimensions. Below, we first describe the shadow teacher preparation during training, then present the three loss components that shape $\hat{\mathbf{g}}_r^i$ during unlearning.

4.3.1 Training-phase Shadow Teacher Preparation

After model stabilization at round t_s, the malicious clients privately train a shadow teacher $\mathbf{w}_{\text{tea}}$ on $\mathcal{D}_{\text{proxy}} \cup \mathcal{D}_{\text{trigger}}$, where $\mathcal{D}_{\text{trigger}}$ is constructed by applying a trigger pattern to samples drawn from $\mathcal{D}_{\text{proxy}}$ and relabeling them to the attacker-chosen target class y_{tar}. The shadow teacher is initialized from $\mathbf{w}_{t_s}$ and optimizes a standard backdoor objective $\mathcal{L}_{\text{tea}}(\mathbf{w}_{\text{tea}}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{\text{proxy}}}[\mathcal{L}(\mathbf{w}_{\text{tea}}; \mathbf{x}, y)] + \gamma_{\text{tea}}\, \mathbb{E}_{(\mathbf{x}',y_{\text{tar}})\sim\mathcal{D}_{\text{trigger}}}[\mathcal{L}(\mathbf{w}_{\text{tea}}; \mathbf{x}', y_{\text{tar}})]$, where γ_{tea} balances clean accuracy and trigger effectiveness.

4.3.2 Unlearning-phase Vacuity Deepening

Although FU induces knowledge vacuity on the masked dimensions, one round may be insufficient for reliable backdoor injection. The attacker therefore participates in unlearning rounds $r = 1, \ldots, R$, receiving the global broadcast model $\tilde{\mathbf{w}}_r$ at each round, with $\tilde{\mathbf{w}}_1 = \mathbf{w}_T$. For $r = 1$, where no prior drift is yet observable, we set $\mathcal{L}_{\text{push}} = 0$. For $r > 1$, rather than pushing parameters in an unconstrained direction, the attacker amplifies the forgetting trajectory by maximizing the alignment between its local update and the observed FU drift $\tilde{\mathbf{w}}_r - \tilde{\mathbf{w}}_{r-1}$ on the masked dimensions:

$$\mathcal{L}_{\text{push}} = -\frac{\left\langle M_{\text{VOID}} \odot \hat{\mathbf{g}}_r^i,\ M_{\text{VOID}} \odot (\tilde{\mathbf{w}}_r - \tilde{\mathbf{w}}_{r-1}) \right\rangle}{\|M_{\text{VOID}} \odot \hat{\mathbf{g}}_r^i\|\, \|M_{\text{VOID}} \odot (\tilde{\mathbf{w}}_r - \tilde{\mathbf{w}}_{r-1})\| + \epsilon}, \tag{10}$$

where $\hat{\mathbf{g}}_r^i$ is the local update of malicious client i at round r and $\odot$ denotes element-wise multiplication. The numerator takes the inner product between the masked malicious update and the observed FU drift, and the denominator normalizes their magnitudes, yielding a negative cosine-similarity loss. Minimizing it aligns the update with the legitimate forgetting direction while ignoring scale.

4.3.3 Backdoor Logic Substitution

While $\mathcal{L}_{\text{push}}$ deepens the vacuity on the masked dimensions, the attacker simultaneously needs to fill them with the desired backdoor behavior. To this end, the attacker distills the trigger-to-target mapping from the shadow teacher via knowledge distillation:

$$\mathcal{L}_{\text{pull}} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{trigger}}^i}\left[\text{KL}\left(\sigma\left(\frac{f(\mathbf{x}; \mathbf{w}_{\text{tea}})}{\tau}\right) \,\middle\|\, \sigma\left(\frac{f(\mathbf{x}; \tilde{\mathbf{w}}_r + \hat{\mathbf{g}}_r^i)}{\tau}\right)\right)\right], \tag{11}$$

where $f(\mathbf{x}; \mathbf{w})$ denotes the model output (logits) at input $\mathbf{x}$ under parameters $\mathbf{w}$, $\mathcal{D}_{\text{trigger}}^i$ is the trigger-embedded samples constructed locally by malicious client i, $\text{KL}(\cdot\|\cdot)$ is the Kullback-Leibler divergence [30], σ is the softmax function, and τ is a distillation temperature that softens the output distributions for more effective knowledge transfer.

4.3.4 Final Crafted Update

While $\mathcal{L}_{\text{push}}$ deepens knowledge vacuity and $\mathcal{L}_{\text{pull}}$ fills the vacated capacity with trigger logic, the crafted update must still remain compatible with the legitimate FU objective. To preserve this consistency, the attacker introduces a retained-compatible unlearning loss:

$$\mathcal{L}_{\text{unlearn}} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{\text{proxy}}}\left[\mathcal{L}(f(\mathbf{x}; \tilde{\mathbf{w}}_r + \hat{\mathbf{g}}_r^i), y)\right] - \mu\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{\text{unlearn}}}\left[\mathcal{L}(f(\mathbf{x}; \tilde{\mathbf{w}}_r + \hat{\mathbf{g}}_r^i), y)\right], \tag{12}$$

where μ controls the strength of the unlearning signal. The full optimization objective is:

$$\mathcal{L}_{\text{VOID}} = \mathcal{L}_{\text{unlearn}} + \alpha\, \mathcal{L}_{\text{push}} + \beta\, \mathcal{L}_{\text{pull}}, \tag{13}$$

where α and β control the strengths of trajectory alignment and trigger distillation, respectively. At each unlearning round $r = 1, \ldots, R$, the attacker optimizes $\mathcal{L}_{\text{VOID}}$ with respect to $\hat{\mathbf{g}}_r^i$ for E_{atk} local steps with masked gradient projection:

$$\hat{\mathbf{g}}_r^i \leftarrow \hat{\mathbf{g}}_r^i - \eta_{\text{atk}}\, M_{\text{VOID}} \odot \nabla_{\hat{\mathbf{g}}_r^i} \mathcal{L}_{\text{VOID}}. \tag{14}$$

The mask M_{VOID} zeroes out gradients on non-masked dimensions, ensuring that only the selected parameters are modified. This projected optimization is equivalent to constraining the crafted update to the writable subspace while leaving non-masked parameters unchanged. The attacker then submits the resulting $\hat{\mathbf{g}}_r^i$ to the server for aggregation. From the server's perspective, $\hat{\mathbf{g}}_r^i$ is aligned with a standard unlearning correction, as the injected trigger semantics stay naturally blended into the legitimate forgetting trajectory over dimensions that FU would modify regardless. Algorithm 1 in Appendix E consolidates the two-phase protocol introduced above.

5 Experiments

5.1 Experimental Setup

Evaluation Setting. We evaluate VOID on four widely used benchmarks spanning image, sensor, and text modalities: MNIST [31], CIFAR-10 [29], Human Activity Recognition (HAR) [3], and AG News [49]. Following standard FL practice, we pair each dataset with a representative model architecture: a lightweight CNN for MNIST, ResNet-18 [23] for CIFAR-10, a HAR-CNN [38] for HAR, and FastText [25] for AG News. We evaluate both IID and non-IID client data partitions, using a Dirichlet distribution for the non-IID setting. We consider three representative calibration-based active FU protocols: FedRR [34], which applies diagonal-Fisher Newton-type corrections; NoT [26], which performs weight negation; and FedAU [21], which incorporates an auxiliary unlearning module during training and performs unlearning through a lightweight linear operation.

Attack Setup. Malicious clients submit clean updates during FL training. For VOID, trigger-embedded samples are used only for private attack preparation, not for FL training updates. Specifically, each malicious client constructs triggered variants from 10% of its local data and relabels them to a fixed target class. Triggers are tailored to each dataset: a 3×3 white square at the bottom-right corner for MNIST [22], a 5×5 red "X" at the bottom-left corner for CIFAR-10 [47], a linearly decreasing sequence in the last four time steps of one channel for HAR [13], and a rare token randomly inserted into the input sentence for AG News [10]. We compare VOID against two representative categories of FU-targeted backdoor attacks: (i) training-stage latent attacks that hide malicious effects during training and activate them by unlearning suppressing data, represented by FUBA [42] and CP-FU [16]; and (ii) unlearning-phase poisoning attacks that directly train a backdoor via trigger-poisoned updates submitted during the unlearning rounds, represented by BadUnlearn [46].

Default Scenario. The default scenario uses MNIST with a lightweight CNN, IID data, FedRR, $n = 100$ clients, $m = 20$ malicious clients, and FedAvg [36] aggregation with 80% of clients randomly per-round. We run 500 FL training rounds followed by $R = 300$ unlearning rounds, and each malicious client requests unlearning for 5% of its local training samples. The remaining optimization, mask-generation, and shadow-teacher hyperparameters are listed in Appendix G (Table 5). Experiments use one RTX 4090 GPU with 24GB memory. Stabilization is declared at round t_s once clean accuracy on $\mathcal{D}_{\text{proxy}}$ stays within tolerance for the prescribed window; t_s varies slightly across seeds and protocols and is used only as the attack preparation point. All results are averaged over three random seeds.

Evaluation Metrics. We evaluate VOID using two standard backdoor metrics [4, 28]: *Attack Success Rate* (ASR), the fraction of triggered inputs classified as the target label y_{tar}, and *Main-task Accuracy* (ACC), the accuracy on clean inputs. Higher ASR indicates stronger attack effectiveness, while higher ACC indicates better clean-task preservation. To quantify durability after the attack ceases, we report the *Persistence Ratio* $\text{PR} = \text{ASR}(t_e + T_p)/\text{ASR}(t_e)$, where t_e is the end of unlearning and $T_p = 100$ denotes benign finetuning rounds under the same client-sampling protocol. Larger PR indicates stronger post-unlearning persistence and is computed only when $\text{ASR}(t_e) \geq 0.6$.

5.2 Experimental Results

Attack Performance across Datasets, Models, and Mask Sparsity. Table 1 reports ASR/ACC across datasets, FU protocols, and mask fraction K, with the no-attack baseline for comparison. VOID is effective across all settings. Once $K \geq 5\%$, ASR exceeds 0.76 in all cases while ACC remains close to the clean baseline. In larger models, the required selected fraction is even smaller: ResNet-18 on CIFAR-10 reaches ASR > 0.79 at $K = 1\%$, and FastText on AG News saturates at $K \geq 3\%$. These results indicate that VOID's effective injection capacity is concentrated in a small high-saliency subspace rather than distributed across the full model.

Impact of Malicious Client Ratio. Figures 2 and 3 report ASR and ACC as the malicious-client ratio m/n varies. VOID achieves non-trivial ASR even at $m/n = 0.1$ and saturates above 98% once $m/n \geq 0.2$. ACC remains above 98% across all attacker ratios, showing that VOID does not affect the main-task performance.

Robustness under Non-IID Client Data. We partition client data using a Dirichlet distribution with parameter α_{Dir}, where smaller values indicate stronger heterogeneity and $\alpha_{\text{Dir}} = 10$ approximates IID.

Table 1: Attack performance of VOID across datasets, models, FU protocols, and mask fraction K. Each entry reports ASR/ACC.

dataset (model)	FU method	w/o attack	Mask fraction K (%)				
			1	3	5	10	20
MNIST (CNN)	FedRR	0.000/0.984	0.216/0.985	0.957/0.984	0.997/0.985	0.998/0.985	0.997/0.987
	NoT	0.000/0.988	0.366/0.967	0.970/0.973	0.999/0.984	0.999/0.983	0.999/0.986
	FedAU	0.000/0.982	0.076/0.979	0.116/0.981	0.999/0.984	0.993/0.988	0.999/0.988
CIFAR-10 (ResNet-18)	FedRR	0.002/0.824	0.935/0.835	0.890/0.853	0.916/0.855	0.920/0.845	0.876/0.832
	NoT	0.003/0.835	0.819/0.837	0.877/0.846	0.899/0.866	0.901/0.868	0.939/0.866
	FedAU	0.003/0.835	0.793/0.828	0.973/0.824	0.978/0.825	0.986/0.837	0.988/0.834
AG News (FastText)	FedRR	0.001/0.899	0.146/0.887	0.883/0.883	0.885/0.885	0.883/0.883	0.825/0.879
	NoT	0.001/0.902	0.354/0.895	0.904/0.894	0.902/0.902	0.899/0.899	0.871/0.894
	FedAU	0.001/0.898	0.382/0.901	0.908/0.898	0.902/0.902	0.901/0.901	0.903/0.899
HAR (HAR-CNN)	FedRR	0.000/0.932	0.183/0.934	0.237/0.926	0.856/0.945	0.898/0.951	0.856/0.933
	NoT	0.001/0.928	0.235/0.938	0.381/0.945	0.792/0.935	0.870/0.943	0.858/0.941
	FedAU	0.000/0.928	0.115/0.929	0.349/0.918	0.762/0.922	0.860/0.921	0.876/0.927

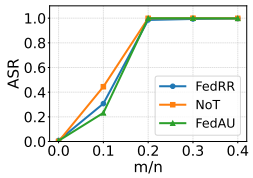

Figure 2: ASR under different m/n.

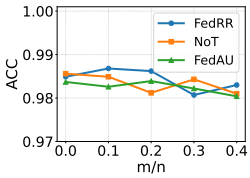

Figure 3: ACC under different m/n.

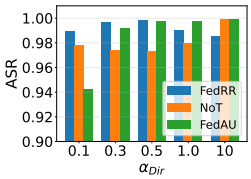

Figure 4: ASR under different non-IID levels.

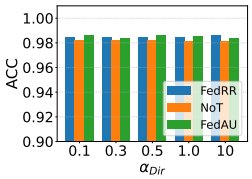

Figure 5: ACC under different non-IID levels.

Figures 4 and 5 show that VOID remains robust across heterogeneity levels. Even at the most skewed setting ($\alpha_{\text{Dir}} = 0.1$), ASR remains above 0.94 across three FU protocols with stable ACC.

Persistence against Post-Unlearning Finetuning and Robust Aggregation. Many FU methods append benign FL rounds after unlearning to stabilize the recovered model and restore utility [52, 26, 8]. Standard FL defenses are unreliable during FU itself because legitimate forgetting-oriented updates may deviate from normal training updates, but robust aggregation can be re-enabled in this finetuning phase. We use the PR to quantify backdoor durability over $T_p = 100$ finetuning rounds under FedAvg and five robust aggregators. Figures 6 and 7 illustrate ASR and ACC from the end of unlearning through post-unlearning finetuning under FedAvg. Round 0 denotes the end of unlearning and measures direct injection performance. At this point, VOID achieves high ASR with ACC close to the clean baseline, while later rounds measure persistence. After unlearning, VOID and BadUnlearn stop injecting malicious updates, so benign finetuning gradually dilutes their backdoors. VOID decays more mildly because its trigger is implanted in the small set of masked parameters that FU is expected to modify most, where retained-data optimization exerts weaker restoring pressure. By contrast, FUBA and CP-FU remain nearly flat because their malicious clients continue reinforcing trigger-embedded updates during finetuning, rather than measuring passive persistence. Table 2 adds robust aggregation during finetuning, with FedAvg serving as the no-defense baseline. Under FedAvg, VOID retains PR of 0.875–0.958, compared with 0.594–0.615 for BadUnlearn, showing stronger passive persistence. VOID and BadUnlearn are largely unaffected by robust aggregation because they stop attacking after unlearning, while FUBA and CP-FU suffer large PR drops under most aggregators as their poisoned updates are filtered. Krum behaves differently because selecting only one update per round weakens the cleansing effect of benign finetuning and yields higher PR across attacks. Overall, VOID remains effective even when defenses are re-enabled.

Table 2: Backdoor persistence under post-unlearning finetuning with robust aggregation. Each entry reports the persistence ratio (PR) after $T_p = 100$ finetuning rounds.

Defense	VOID			BadUnlearn			FUBA			CP-FU		
	FedRR	NoT	FedAU	FedRR	NoT	FedAU	FedRR	NoT	FedAU	FedRR	NoT	FedAU
FedAvg	0.875	0.958	0.890	0.594	0.615	0.615	1.051	0.999	1.000	1.012	1.000	1.000
Median	0.884	0.958	0.886	0.586	0.631	0.615	0.582	0.698	0.612	0.571	0.622	0.643
Trimmed	0.876	0.958	0.889	0.602	0.626	0.608	0.568	0.713	0.613	0.553	0.615	0.654
Krum	0.951	0.962	0.948	0.828	0.682	0.690	0.703	0.792	0.723	0.725	0.822	0.790
Multi-Krum	0.864	0.957	0.891	0.708	0.632	0.594	0.579	0.721	0.621	0.561	0.630	0.648
Flame	0.878	0.959	0.885	0.660	0.634	0.617	0.580	0.726	0.622	0.572	0.608	0.654

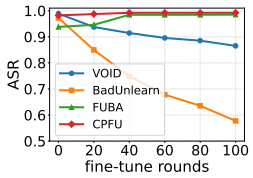

Figure 6: ASR over post-unlearning finetuning rounds.

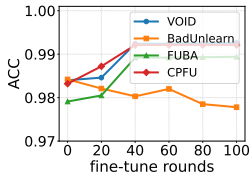

Figure 7: ACC over post-unlearning finetuning rounds.

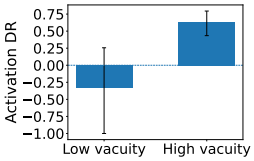

Figure 8: Empirical evidence of knowledge vacuity.

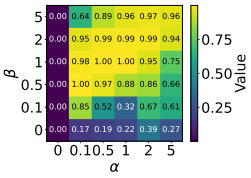

Figure 9: ASR heatmap over the push weight α and pull weight β.

Empirical Evidence of Knowledge Vacuity. To verify that the vacuity score captures FU-weakened representations independently of the attack, we measure neuron-level activation changes before and after a no-attack FU process. Neurons are ranked by their pre-unlearning vacuity scores, and the bottom and top 10% form the low-vacuity and high-vacuity groups, respectively. For each neuron a, we compute the activation drop ratio (DR) as $\mathrm{Drop}(a) = (\bar{a}_{\mathrm{pre}}^{\mathrm{unlearn}} - \bar{a}_{\mathrm{post}}^{\mathrm{unlearn}})/(\bar{a}_{\mathrm{pre}}^{\mathrm{unlearn}} + \epsilon)$, where $\bar{a}_{\mathrm{pre}}^{\mathrm{unlearn}}$ and $\bar{a}_{\mathrm{post}}^{\mathrm{unlearn}}$ are the mean absolute activations on $\mathcal{D}_{\mathrm{unlearn}}$ before and after FU. Figure 8 reports the median DR with 95% bootstrap confidence intervals. High-vacuity neurons exhibit a substantially larger median DR than low-vacuity neurons, indicating stronger FU-induced contraction. In contrast, the low-vacuity group has a median DR below zero and a wider confidence interval crossing zero, showing no stable contraction pattern. This separation suggests that the vacuity score identifies representations that are preferentially affected by FU, rather than an artifact of the attack.

5.3 Ablation Studies

Effect of Loss Weights α and β. Figure 9 plots ASR under different loss weights α and β. Setting $\alpha = 0$ makes ASR collapse to 0, showing that the push term is indispensable for activating the backdoor objective. Setting $\beta = 0$ also limits ASR to at most 0.391, indicating that the pull term is needed to stabilize the trigger feature during unlearning. When both terms are enabled, VOID achieves ASR above 0.85 over a broad central region. Thus, VOID is robust to hyperparameter choices and only degrades under extreme imbalance between the two loss terms.

Contribution of Influence Selection and Mask Restriction. We compare the VOID mask with four variants in Table 3. The *w/o mask* variant applies dense updates, while *influence-only*, *vacuity-only*, and *random mask* use the same selected-parameter fraction K as VOID but rank coordinates by influence, vacuity, or random sampling. We show ASR/ACC at the end of unlearning and after $T_p = 100$ clean finetuning rounds under three FU protocols. The results show that the influence signal is critical for successful injection. *Vacuity-only* and *random mask* use the same selected fraction as VOID but are poorly aligned with FU-driven directions, causing ASR after unlearning to fall below 0.30. In contrast, *influence-only* and the full VOID mask reach ASR close to 1.0. Restricting the update to a small selected subset is also critical for persistence. The dense *w/o mask* variant achieves high ASR after unlearning, but drops to about 0.60 after clean finetuning because the backdoor is spread across all parameters. By concentrating the trigger in the selected subset, *influence-only* and the full VOID mask retain ASR above 0.88 across all FU protocols. These results show that influence selection drives injection and mask restriction improves persistence. Vacuity profiling alone is insufficient, but when combined with influence selection, it serves as a compatibility-oriented filter that biases the mask toward representations disproportionately shaped by the removed data.

Table 3: ASR/ACC across five mask designs and three FU protocols, in the unlearning phase and post-unlearning finetuning phase.

FU method	Phase	w/o mask	Influence-only	Vacuity-only	Random mask	VOID mask
FedRR	unlearn	0.999/0.991	0.999/0.985	0.156/0.986	0.298/0.967	0.997/0.985
	fine-tune	0.595/0.987	0.925/0.986	0.092/0.989	0.041/0.977	0.888/0.985
NoT	unlearn	0.982/0.984	0.999/0.982	0.163/0.988	0.272/0.971	0.999/0.984
	fine-tune	0.603/0.987	0.933/0.985	0.129/0.985	0.053/0.979	0.957/0.984
FedAU	unlearn	0.989/0.987	1.000/0.984	0.163/0.979	0.275/0.966	0.999/0.984
	fine-tune	0.608/0.988	0.934/0.985	0.116/0.982	0.055/0.975	0.889/0.985

Additional evaluations on runtime overhead, the unlearn data ratio, distillation temperature, and LISSA hyperparameters are provided in Appendix H.

6 Conclusion and Future Work

In this paper, we identify knowledge vacuity as a structural vulnerability of approximate FU and propose VOID, an unlearning-phase backdoor attack that uses the forgetting trajectory to carry concealed trigger semantics. Experiments show high attack success, preserved clean accuracy, and post-unlearning persistence. By exposing a dual-use FU security risk, this work motivates defenses that distinguish legitimate forgetting from injection-aligned updates.

Acknowledgments

This work was partially supported by NSF under grants 2319781 and 2505381.

References

[1] Naman Agarwal, Brian Bullins, and Elad Hazan. Second-order stochastic optimization for machine learning in linear time. *Journal of Machine Learning Research*, 18(116):1–40, 2017.

[2] Sebastien Andreina, Giorgia Azzurra Marson, Helen Möllering, and Ghassan Karame. BAFFLE: Backdoor detection via feedback-based federated learning. In *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 852–863, 2021.

[3] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2013.

[4] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2938–2948, 2020.

[5] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[6] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 634–643, 2019.

[7] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[8] Xiaoyu Cao, Jinyuan Jia, Zaixi Zhang, and Neil Zhenqiang Gong. FedRecover: Recovering from poisoning attacks in federated learning using historical information. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, pages 1366–1383, 2023.

[9] Tianshi Che, Yang Zhou, Zijie Zhang, Lingjuan Lyu, Ji Liu, Da Yan, Dejing Dou, and Jun Huan. Fast federated machine unlearning with nonlinear functional theory. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4241–4268, 2023.

[10] Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. BadNL: Backdoor attacks against NLP models with semantic-preserving improvements. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, pages 554–569, 2021.

[11] Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25, 2017.

[12] Yanbo Dai and Songze Li. Chameleon: Adapting to peer images for planting durable backdoors in federated learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 6712–6725, 2023.

[13] Daizong Ding, Mi Zhang, Yuanmin Huang, Xudong Pan, Fuli Feng, Erling Jiang, and Min Yang. Towards backdoor attack on deep learning based time series classification. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pages 1274–1287, 2022.

[14] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to Byzantine-Robust federated learning. In *Proceedings of the USENIX Security Symposium (USENIX Security)*, pages 1605–1622, 2020.

[15] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *Proceedings of the International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, pages 301–316, 2020.

[16] Kun Gao, Tianqing Zhu, Dayong Ye, Bo Liu, and Wanlei Zhou. Hidden threats in federated unlearning: Camouflaged poisoning attacks and their unlearning consequences. *IEEE Transactions on Dependable and Secure Computing*, 2025.

[17] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9304–9312, 2020.

[18] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 383–398, 2020.

[19] Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 792–801, 2021.

[20] Xueluan Gong, Yanjiao Chen, Huayang Huang, Yuqing Liao, Shuai Wang, and Qian Wang. Coordinated backdoor attacks against federated learning with model-dependent triggers. *IEEE Network*, 36(1):84–90, 2022.

[21] Hanlin Gu, Gongxi Zhu, Jie Zhang, Xinyuan Zhao, Yuxing Han, Lixin Fan, and Qiang Yang. Unlearning during learning: An efficient federated machine unlearning method. *arXiv preprint arXiv:2405.15474*, 2024.

[22] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[24] Yu Jiang, Jiyuan Shen, Ziyao Liu, Chee Wei Tan, and Kwok-Yan Lam. Towards efficient and certified recovery from poisoning attacks in federated learning. *IEEE Transactions on Information Forensics and Security*, 2025.

[25] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomáš Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 427–431, 2017.

[26] Yasser H. Khalil, Leo Brunswic, Soufiane Lamghari, Xu Li, Mahdi Beitollahi, and Xi Chen. NoT: Federated unlearning via weight negation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25759–25769, 2025.

[27] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1885–1894, 2017.

[28] Torsten Krauß and Alexandra Dmitrienko. MESAS: Poisoning defense for federated learning resilient against adaptive attackers. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1526–1540, 2023.

[29] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[30] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[32] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

[33] Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. FedEraser: Enabling efficient client-level data removal from federated learning models. In *Proceedings of the IEEE/ACM International Symposium on Quality of Service (IWQoS)*, pages 1–10, 2021.

[34] Yi Liu, Lei Xu, Xingliang Yuan, Cong Wang, and Bo Li. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 1749–1758, 2022.

[35] Ziyao Liu, Yu Jiang, Jiyuan Shen, Minyi Peng, Kwok-Yan Lam, Xingliang Yuan, and Xiaoning Liu. A survey on federated unlearning: Challenges, methods, and future directions. *ACM Computing Surveys*, 57(1):1–38, 2024.

[36] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282, 2017.

[37] Syed Irfan Ali Meerza, Amir Sadovnik, and Jian Liu. Confuse: Confusion-based federated unlearning with salience exploration. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 427–432, 2024.

[38] Ronald Mutegeki and Dong Seog Han. A CNN-LSTM approach to human activity recognition. In *Proceedings of the International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 362–366, 2020.

[39] Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, et al. FLAME: Taming backdoors in federated learning. In *Proceedings of the USENIX Security Symposium (USENIX Security)*, pages 1415–1432, 2022.

[40] Phillip Rieger, Thien Duc Nguyen, Markus Miettinen, and Ahmad-Reza Sadeghi. Deepsight: Mitigating backdoor attacks in federated learning through deep model inspection. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2022.

[41] Nicol N Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14(7):1723–1738, 2002.

[42] Xinyi Sheng, Wei Bao, Yichen Guo, and Sen Fu. FUBA: Backdoor federated learning via federated unlearning. *IEEE Transactions on Artificial Intelligence*, 2025.

[43] Ningxin Su and Baochun Li. Asynchronous federated unlearning. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 1–10, 2023.

[44] Paul Voigt and Axel Von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer International Publishing, Cham, 1st edition, 2017.

[45] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 16070–16084, 2020.

[46] Wenbin Wang, Qiwen Ma, Zifan Zhang, Yuchen Liu, Zhuqing Liu, and Minghong Fang. Poisoning attacks and defenses to federated unlearning. In *Companion Proceedings of the ACM Web Conference (WWW Companion)*, pages 1365–1369, 2025.

[47] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. DBA: Distributed backdoor attacks against federated learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[48] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 5650–5659, 2018.

[49] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

[50] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 2545–2555, 2022.

[51] Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael Mahoney, Prateek Mittal, Ramchandran Kannan, and Joseph Gonzalez. Neurotoxin: Durable backdoors in federated learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 26429–26446, 2022.

[52] Wenwei Zhao, Xiaowen Li, Yao Liu, and Zhuo Lu. Adversarial update-based federated unlearning for poisoned model recovery. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1071–1075, 2026.

[53] Wenwei Zhao, Yuanzhe Peng, Xiaowen Li, Jie Xu, Yao Liu, and Zhuo Lu. Malicious forgetting: Backdoor injection in active federated unlearning and countermeasure design. In *IEEE INFOCOM 2026-IEEE Conference on Computer Communications*, pages 1–10, 2026.

Appendix

A Extended Related Work: FL Backdoor Attacks, Defenses, and Federated Unlearning

This appendix expands the compact survey in Sections 2.1 and 2.2 with a broader discussion of training-phase backdoor attacks, defenses, and federated unlearning methods.

Backdoor attacks in FL. Backdoor attacks in FL aim to implant a hidden trigger-to-target mapping into the global model while preserving clean-task performance. Early work showed that local model poisoning with trigger-embedded data can manipulate the aggregated model through scaled malicious updates [4, 6]. Subsequent works have explored a variety of strategies to enhance stealth and durability, including distributed trigger decomposition across colluding clients [47, 20], edge-case poisoning in low-density regions of the benign distribution [45], defense-aware gradient manipulation that steers updates around robust aggregators [5, 14], and durability-oriented coordinate selection that confines poisoning to parameters with small benign gradient magnitude [51, 12]. These attacks all operate within a stable training loop and do not exploit the dynamics introduced when parts of the training signal are deliberately removed.

Defenses against FL backdoor attacks. Existing defenses against FL backdoor attacks largely fall into two categories: robust aggregation and anomaly detection. Robust aggregation replaces plain averaging with outlier-resistant estimators, assuming benign updates cluster tightly around the true training gradient. Representative methods include Krum and Multi-Krum, which select updates minimizing pairwise distance to their neighbors [7], coordinate-wise median and trimmed mean, which apply robust statistics independently per parameter [48, 14, 11], and FoolsGold, which penalizes clients whose updates exhibit unusually high pairwise similarity [15]. Anomaly detection

instead identifies malicious participants by their behavioral deviations, either through clustering and dynamic norm bounding of per-round updates [39, 40] or by tracking multi-round update trajectories to flag clients that drift from the empirical benign distribution [50, 2]. Both categories are designed and evaluated under training-phase assumptions, with no mechanism specifically tailored to the unlearning phase.

Federated unlearning methods. Federated unlearning removes the influence of a designated subset of training data from a trained global model while preserving utility on the retained data. Existing approaches fall into two broad families. Reconstruction-based methods resume training from saved intermediate states or stored historical updates: FedEraser [33] reconstructs the unlearned model by subtracting forgotten-client contributions from calibrated historical updates, FedRecover [8] replays stored gradients to recover from poisoning, and rapid-retraining schemes [34] accelerate exact retraining with warm-start initializations. These methods approximate exact retraining closely but incur substantial storage and communication overhead for maintaining per-round update histories. Calibration-based methods instead start from the converged model and apply finite-step post-hoc corrections, including weight-negation schemes that invert parameter directions most coupled to the forgotten data [26], certified recovery procedures that bound the residual influence of poisoned clients [24], and asynchronous unlearning tailored to staggered client availability [43]. A recent survey [35] provides a broader taxonomy and systematic comparison. In practice, calibration-based methods dominate deployments due to their efficiency, at the cost of residual approximation errors concentrated on dimensions most influenced by the forgotten data.

B Knowledge Vacuity in Federated Unlearning: Full Derivation

This appendix provides the formal proofs supporting Section 4.1. We show that (1) the FU correction concentrates its energy on the forgotten-data footprint, and (2) perturbations implanted on these dimensions persist through subsequent retained-data optimization.

B.1 Derivation of the Approximate FU Correction

Let $\mathbf{w}_T$ denote the converged global model trained on $\mathcal{D} = \mathcal{D}_{\text{retain}} \cup \mathcal{D}_{\text{unlearn}}$. By optimality, $\mathbf{w}_T$ satisfies the first-order condition

$$\nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D};\, \mathbf{w}_T) = \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{retain}};\, \mathbf{w}_T) + \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{unlearn}};\, \mathbf{w}_T) = \mathbf{0}. \tag{15}$$

The ideal unlearned model $\mathbf{w}^*$ minimizes only the retained loss, so it satisfies $\nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{retain}};\, \mathbf{w}^*) = \mathbf{0}$. Expanding around $\mathbf{w}_T$ via a second-order Taylor approximation:

$$\nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{retain}};\, \mathbf{w}^*) \approx \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{retain}};\, \mathbf{w}_T) + H\,(\mathbf{w}^* - \mathbf{w}_T) = \mathbf{0}, \tag{16}$$

where $H = \nabla^2_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{retain}};\, \mathbf{w}_T)$ is the Hessian of the retained loss. Substituting Eq. (15) gives $\nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{retain}};\, \mathbf{w}_T) = -\nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{unlearn}};\, \mathbf{w}_T)$, so:

$$\Delta \mathbf{w}_{\text{FU}} = \mathbf{w}^* - \mathbf{w}_T \approx H^{-1}\, \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{unlearn}};\, \mathbf{w}_T). \tag{17}$$

This is the Newton step for removing $\mathcal{D}_{\text{unlearn}}$'s influence, and it coincides with the classical influence-function estimator [27].

B.2 Footprint–Update Energy Concentration

We now show that the energy of $\Delta \mathbf{w}_{\text{FU}}$ concentrates on the forgotten-data footprint. Let $\{(\lambda_i, \mathbf{e}_i)\}_{i=1}^d$ be the eigen-decomposition of H, with eigenvalues ordered $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d > 0$. Decomposing the gradient $\mathbf{g} = \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{unlearn}};\, \mathbf{w}_T) = \sum_i g_i \mathbf{e}_i$, the FU correction becomes:

$$\Delta \mathbf{w}_{\text{FU}} = \sum_{i=1}^{d} \frac{g_i}{\lambda_i}\, \mathbf{e}_i. \tag{18}$$

Definition 1 (Forgotten-data footprint)**.** *The forgotten-data footprint $\mathcal{F}_\rho$ is the minimal index set such that $\sum_{i \in \mathcal{F}_\rho} g_i^2 \geq \rho \|\mathbf{g}\|^2$ for a chosen concentration threshold $\rho \in (0, 1]$.*

Proposition 1 (Footprint–update alignment)**.** *The fraction of update energy carried by $\mathcal{F}_\rho$ satisfies*

$$\frac{\sum_{i\in\mathcal{F}_\rho} g_i^2/\lambda_i^2}{\sum_{i=1}^d g_i^2/\lambda_i^2} \;\geq\; \rho\cdot\frac{\lambda_{\min}(\mathcal{F}_\rho^c)^2}{\lambda_{\max}(\mathcal{F}_\rho)^2+\lambda_{\min}(\mathcal{F}_\rho^c)^2}, \tag{19}$$

where $\lambda_{\min}(\mathcal{F}_\rho^c)=\min_{i\notin\mathcal{F}_\rho}\lambda_i$ and $\lambda_{\max}(\mathcal{F}_\rho)=\max_{i\in\mathcal{F}_\rho}\lambda_i$. In particular, when the footprint dimensions have low curvature (λ_i small for $i\in\mathcal{F}_\rho$), the $1/\lambda_i^2$ amplification further concentrates update energy on $\mathcal{F}_\rho$ beyond the gradient concentration alone.

Proof. From Eq. (18), the squared update norm decomposes as $\|\Delta\mathbf{w}_{\text{FU}}\|^2=\sum_i g_i^2/\lambda_i^2$. For $i\in\mathcal{F}_\rho$, the contribution is $g_i^2/\lambda_i^2\geq g_i^2/\lambda_{\max}(\mathcal{F}_\rho)^2$. For $i\notin\mathcal{F}_\rho$, the contribution is $g_i^2/\lambda_i^2\leq g_i^2/\lambda_{\min}(\mathcal{F}_\rho^c)^2$. Therefore:

$$\begin{aligned}\frac{\sum_{i\in\mathcal{F}_\rho} g_i^2/\lambda_i^2}{\sum_i g_i^2/\lambda_i^2} &\geq \frac{\sum_{i\in\mathcal{F}_\rho} g_i^2/\lambda_{\max}(\mathcal{F}_\rho)^2}{\sum_{i\in\mathcal{F}_\rho} g_i^2/\lambda_{\max}(\mathcal{F}_\rho)^2+\sum_{i\notin\mathcal{F}_\rho} g_i^2/\lambda_{\min}(\mathcal{F}_\rho^c)^2}\\ &= \frac{S_\rho/\lambda_{\max}(\mathcal{F}_\rho)^2}{S_\rho/\lambda_{\max}(\mathcal{F}_\rho)^2+(1-\rho)\|\mathbf{g}\|^2/\lambda_{\min}(\mathcal{F}_\rho^c)^2},\end{aligned} \tag{20}$$

where $S_\rho=\sum_{i\in\mathcal{F}_\rho} g_i^2\geq\rho\|\mathbf{g}\|^2$. Substituting and simplifying yields the stated bound. □

Implication. Since the footprint dimensions correspond to parameters specialized for $\mathcal{D}_{\text{unlearn}}$, they tend to have low λ_i (see next subsection). The $1/\lambda_i^2$ weighting in Eq. (18) amplifies their contribution, so $\Delta\mathbf{w}_{\text{FU}}$ is dominated by changes on the footprint. From the server's perspective, these large changes are the expected outcome of unlearning, and additional perturbations on the same dimensions naturally blend into the legitimate correction.

B.3 Low-Curvature Stability of Implanted Perturbations

We now show that perturbations implanted on the footprint dimensions persist through subsequent retained-data optimization.

Proposition 2 (Implant stability)**.** *Let $\mathbf{w}_{FU}$ be the model after unlearning and backdoor injection. Suppose the attacker's perturbation $\boldsymbol{\delta}$ is supported on the footprint $\mathcal{F}_\rho$, i.e., $\mathbf{w}_{FU}=\mathbf{w}^*+\boldsymbol{\delta}$ with $\delta_i=0$ for $i\notin\mathcal{F}_\rho$. Consider Q rounds of gradient descent on $\mathcal{D}_{retain}$ with learning rate η:*

$$\mathbf{w}_{k+1}=\mathbf{w}_k-\eta\,\nabla_{\mathbf{w}}\mathcal{L}(\mathcal{D}_{retain};\,\mathbf{w}_k),\quad \mathbf{w}_0=\mathbf{w}_{FU}. \tag{21}$$

Under a local quadratic approximation of $\mathcal{L}(\mathcal{D}_{retain})$ around $\mathbf{w}^*$, the residual perturbation after Q steps satisfies, in the i-th eigendirection of H:

$$|\delta_i^{(Q)}|=|1-\eta\lambda_i|^Q\,|\delta_i|,\quad i\in\mathcal{F}_\rho. \tag{22}$$

Proof. Under the quadratic approximation $\mathcal{L}(\mathcal{D}_{\text{retain}};\,\mathbf{w})\approx\mathcal{L}(\mathcal{D}_{\text{retain}};\,\mathbf{w}^*)+\frac{1}{2}(\mathbf{w}-\mathbf{w}^*)^\top H\,(\mathbf{w}-\mathbf{w}^*)$, the gradient at $\mathbf{w}_k$ is $\nabla\mathcal{L}=H(\mathbf{w}_k-\mathbf{w}^*)$. Let $\boldsymbol{\delta}^{(k)}=\mathbf{w}_k-\mathbf{w}^*$. Then $\boldsymbol{\delta}^{(k+1)}=(I-\eta H)\boldsymbol{\delta}^{(k)}$. In the eigenbasis of H, each component evolves independently: $\delta_i^{(k+1)}=(1-\eta\lambda_i)\,\delta_i^{(k)}$, yielding $\delta_i^{(Q)}=(1-\eta\lambda_i)^Q\,\delta_i$. □

Implication. For footprint dimensions where $\lambda_i\ll 1$, the decay factor $(1-\eta\lambda_i)^Q\approx 1-Q\eta\lambda_i\approx 1$ even after many rounds Q. Concretely, the number of rounds required to halve the perturbation on dimension i is:

$$Q_{1/2}(i)=\frac{\ln 2}{\ln(1/(1-\eta\lambda_i))}\approx\frac{\ln 2}{\eta\lambda_i}. \tag{23}$$

For a dimension with $\lambda_i=10^{-3}$ and $\eta=0.01$, this gives $Q_{1/2}\approx 69{,}315$ rounds, far beyond any practical FL deployment. The implanted perturbation is therefore a near-stable fixed point of retained-data optimization, with little gradient pressure to reclaim its original semantic role.

B.4 Combined Result

Propositions 1 and 2 together establish the main claim. The forgotten-data footprint is simultaneously the subspace where (a) the FU correction concentrates its energy, allowing injected perturbations to blend into the legitimate update, and (b) the retained-data curvature is smallest, allowing the implanted semantics to persist. This follows directly from the spectral structure of Eq. (17), where the same small eigenvalues λ_i that force FU to make large corrections (g_i/λ_i) also leave little optimization pressure for subsequent training to reclaim these dimensions (decay $\approx (1-\eta\lambda_i)^Q \approx 1$). FU therefore creates the knowledge vacuity exploited by VOID as a geometric byproduct of its own operation.

C Derivation of the LISSA Recursion

In this appendix, we provide the full derivation showing that the LISSA recursion converges to the inverse Hessian-vector product used in Section 4.2.1.

Setup. Let $H = H(\mathbf{w}_{t_s})$ denote the Hessian of the loss on $\mathcal{D}_{\text{proxy}}$ at stabilization, used as a proxy for the retained-data Hessian, and let $\mathbf{b} = \nabla_{\mathbf{w}}\mathcal{L}(\mathcal{D}_{\text{unlearn}}; \mathbf{w}_{t_s})$. Our goal is to compute $(H+\lambda_{\text{LISSA}}I)^{-1}\mathbf{b}$ without explicitly forming or inverting the $d \times d$ matrix $H + \lambda_{\text{LISSA}}I$. This quantity has the same direction as $\mathbf{v}_{\text{inf}}$ in Section 4.2.1. Throughout, $\eta_{\text{L}} > 0$ denotes the LISSA step size used in the main-text recursion (Eq. (5)), chosen so that $\eta_{\text{L}}(\lambda_{\max}(H) + \lambda_{\text{LISSA}}) < 2$.

Neumann series expansion. The step-size condition above guarantees that all eigenvalues of $\eta_{\text{L}}(H + \lambda_{\text{LISSA}}I)$ lie in $(0, 2)$, so $\rho\big(I - \eta_{\text{L}}(H + \lambda_{\text{LISSA}}I)\big) < 1$ and the matrix $\eta_{\text{L}}(H + \lambda_{\text{LISSA}}I)$ admits a Neumann-series inverse:

$$\big(\eta_{\text{L}}(H + \lambda_{\text{LISSA}}I)\big)^{-1} = \sum_{k=0}^{\infty} \big(I - \eta_{\text{L}}(H + \lambda_{\text{LISSA}}I)\big)^k. \tag{24}$$

Applying both sides to $\mathbf{b}$ and rearranging gives

$$(H + \lambda_{\text{LISSA}}I)^{-1}\mathbf{b} = \eta_{\text{L}} \sum_{k=0}^{\infty} \big(I - \eta_{\text{L}}(H + \lambda_{\text{LISSA}}I)\big)^k \mathbf{b}. \tag{25}$$

Under a generalized Gauss–Newton PSD curvature surrogate [41], $\lambda_{\text{LISSA}} > 0$ shifts the spectrum away from zero. If the raw Hessian is used instead, we assume the damping is chosen so that $H + \lambda_{\text{LISSA}}I$ is positive definite, after which the step-size condition above can be satisfied by choosing η_{L} small enough.

Recursive formulation. Define the partial sum $\mathbf{h}_j = \sum_{k=0}^{j} \big(I - \eta_{\text{L}}(H + \lambda_{\text{LISSA}}I)\big)^k \mathbf{b}$. Then:

$$\mathbf{h}_0 = \mathbf{b}, \tag{26}$$

$$\mathbf{h}_j = \mathbf{b} + \big(I - \eta_{\text{L}}(H + \lambda_{\text{LISSA}}I)\big)\,\mathbf{h}_{j-1}, \quad j = 1, 2, \ldots \tag{27}$$

This can be verified by induction. For $j = 1$: $\mathbf{h}_1 = \mathbf{b} + (I - \eta_{\text{L}}(H + \lambda_{\text{LISSA}}I))\mathbf{b} = \sum_{k=0}^{1}(I - \eta_{\text{L}}(H + \lambda_{\text{LISSA}}I))^k\mathbf{b}$. Assuming $\mathbf{h}_{j-1} = \sum_{k=0}^{j-1}(I - \eta_{\text{L}}(H + \lambda_{\text{LISSA}}I))^k\mathbf{b}$, we have:

$$\begin{aligned}
\mathbf{h}_j &= \mathbf{b} + (I - \eta_{\text{L}}(H + \lambda_{\text{LISSA}}I)) \sum_{k=0}^{j-1}(I - \eta_{\text{L}}(H + \lambda_{\text{LISSA}}I))^k\mathbf{b} \\
&= \mathbf{b} + \sum_{k=1}^{j}(I - \eta_{\text{L}}(H + \lambda_{\text{LISSA}}I))^k\mathbf{b} \\
&= \sum_{k=0}^{j}(I - \eta_{\text{L}}(H + \lambda_{\text{LISSA}}I))^k\mathbf{b}.
\end{aligned} \tag{28}$$

As $j \to \infty$, this partial sum converges to the geometric series in Eq. (25) (without the leading η_{L} factor), yielding $\mathbf{h}_\infty = \frac{1}{\eta_{\text{L}}}(H + \lambda_{\text{LISSA}}I)^{-1}\mathbf{b}$. That is, the iterate differs from the target inverse-Hessian-vector product by a known global scalar $1/\eta_{\text{L}}$. Because the trajectory signal $|(\mathbf{v}_{\text{inf}})_d|$ is subsequently min-max normalized in the saliency fusion step (Eq. (8)), this constant factor cancels

and does not affect the resulting mask. We therefore use $\mathbf{h}_J$ directly as the trajectory signal in the main text.

Stochastic approximation. In practice, computing the exact Hessian H requires a full pass over the dataset, which is expensive. LISSA replaces H with a stochastic estimate. At each iteration j, a mini-batch z_j is sampled uniformly from $\mathcal{D}_{\text{proxy}}$, and the recursion becomes:

$$\mathbf{h}_j = \mathbf{b} + \left(I - \eta_{\text{L}}(\nabla^2 \mathcal{L}(z_j) + \lambda_{\text{LISSA}} I)\right) \mathbf{h}_{j-1}, \tag{29}$$

which matches Eq. (5) of the main text. Since $\mathbb{E}_{z_j}[\nabla^2 \mathcal{L}(z_j)] = H$, this is an unbiased stochastic approximation of the deterministic recursion. Under standard assumptions (bounded spectral norm of individual Hessians and sufficient damping), the stochastic iterates $\mathbf{h}_j$ converge in expectation to $\frac{1}{\eta_{\text{L}}}(H + \lambda_{\text{LISSA}} I)^{-1}\mathbf{b}$ [1].

Computational cost. Each iteration requires one Hessian-vector product $\nabla^2 \mathcal{L}(z_j)\, \mathbf{h}_{j-1}$, which can be computed in $O(d)$ time via reverse-mode automatic differentiation without ever forming the full Hessian matrix. Over J iterations, the total cost is $O(Jd)$, compared to $O(d^3)$ for explicit inversion. In our experiments, we find that $J = 10$ to 20 iterations suffice for convergence in typical FL model sizes.

D Manifold Contraction in Federated Unlearning

In Section 4.2.2, we claim that FU weakens the semantic role of neurons predominantly shaped by $\mathcal{D}_{\text{unlearn}}$, a phenomenon we term *manifold contraction*. Here we provide a theoretical justification for this phenomenon.

D.1 Theoretical Analysis

Consider a neuron a in an intermediate layer with pre-activation $a(\mathbf{x}) = \mathbf{w}_a^\top \phi_{\ell-1}(\mathbf{x})$, where $\mathbf{w}_a$ is the weight vector and $\phi_{\ell-1}(\mathbf{x})$ is the input from the preceding layer. Suppose that before unlearning, the neuron is predominantly shaped by $\mathcal{D}_{\text{unlearn}}$ in the sense that:

$$\mathbb{E}_{\mathbf{x}\in\mathcal{D}_{\text{unlearn}}}[a(\mathbf{x})] \gg \mathbb{E}_{\mathbf{x}\in\mathcal{D}_{\text{proxy}}}[a(\mathbf{x})]. \tag{30}$$

Effect of gradient-based unlearning. FU algorithms typically update the model by applying a correction of the form $\Delta\mathbf{w} \propto -\nabla_{\mathbf{w}}\mathcal{L}(\mathcal{D}_{\text{unlearn}}; \mathbf{w})$. For the neuron weight $\mathbf{w}_a$, this gradient is dominated by the activations on $\mathcal{D}_{\text{unlearn}}$:

$$\Delta\mathbf{w}_a \propto -\mathbb{E}_{\mathbf{x}\in\mathcal{D}_{\text{unlearn}}}\left[\frac{\partial \mathcal{L}}{\partial a(\mathbf{x})} \cdot \phi_{\ell-1}(\mathbf{x})\right]. \tag{31}$$

Since the neuron is highly activated by $\mathcal{D}_{\text{unlearn}}$, the gradient magnitude $\|\Delta\mathbf{w}_a\|$ is large, leading to a substantial reduction in $a(\mathbf{x})$ for $\mathbf{x} \in \mathcal{D}_{\text{unlearn}}$.

Asymmetric impact on benign data. Crucially, because $\mathbf{w}_a$ contributes disproportionately less to the representation of $\mathcal{D}_{\text{proxy}}$ (by the assumption above), the same update $\Delta\mathbf{w}_a$ has a negligible effect on $\mathbb{E}_{\mathbf{x}\in\mathcal{D}_{\text{proxy}}}[a(\mathbf{x})]$. The post-unlearning activation profile therefore exhibits a sharp asymmetry:

$$\underbrace{\mathbb{E}_{\mathbf{x}\in\mathcal{D}_{\text{unlearn}}}[a'(\mathbf{x})]}_{\text{drops sharply}} \ll \underbrace{\mathbb{E}_{\mathbf{x}\in\mathcal{D}_{\text{unlearn}}}[a(\mathbf{x})]}_{\text{pre-unlearning}}, \qquad \underbrace{\mathbb{E}_{\mathbf{x}\in\mathcal{D}_{\text{proxy}}}[a'(\mathbf{x})]}_{\text{nearly unchanged}} \approx \underbrace{\mathbb{E}_{\mathbf{x}\in\mathcal{D}_{\text{proxy}}}[a(\mathbf{x})]}_{\text{pre-unlearning}}, \tag{32}$$

where a' denotes the neuron activation after unlearning. This is the manifold contraction. The representational capacity allocated to $\mathcal{D}_{\text{unlearn}}$ is weakened, leaving these dimensions in a weakly-constrained state susceptible to modification.

Connection to Hessian geometry. The neurons most susceptible to manifold contraction are precisely those in low-curvature directions of the Hessian. For these dimensions, the loss surface is flat with respect to $\mathcal{D}_{\text{proxy}}$, meaning the model's performance on retained data is insensitive to parameter changes. FU can therefore aggressively modify these dimensions to erase $\mathcal{D}_{\text{unlearn}}$'s influence without degrading main-task accuracy. This insensitivity also means that the weakened dimensions are unlikely to be quickly reclaimed by subsequent training on $\mathcal{D}_{\text{retain}}$, as the gradient signal from retained data along these directions is relatively small.

E Full VOID Attack Algorithm

This appendix summarizes the two-phase VOID attack (Section 4) and its equations.

Algorithm 1 VOID Attack: collective protocol over malicious clients $\mathcal{C}_{\text{mal}}$.

Require: Per-client data $\{\mathcal{D}_i\}_{i\in\mathcal{C}_{\text{mal}}}$, unlearning subset $\mathcal{D}_{\text{unlearn}} \subset \bigcup_{i\in\mathcal{C}_{\text{mal}}} \mathcal{D}_i$, trigger pattern, target class y_{tar}; LISSA hyperparameters $(\eta_{\text{L}}, \lambda_{\text{LISSA}}, J)$; attack hyperparameters $(K, \tau, \mu, \alpha, \beta, \gamma_{\text{tea}}, \eta_{\text{atk}}, R, E_{\text{atk}})$

Ensure: Trigger-implanted global model after federated unlearning

Phase 1: Training

for FL round $t = 1, 2, \ldots$ **do**
  each $i \in \mathcal{C}_{\text{mal}}$ submits a clean local update; monitor global model
  if global model has stabilized **then**
    $t_s \leftarrow t$; **break**
  end if
end for
receive broadcast model $\mathbf{w}_{t_s}$; form $\mathcal{D}_{\text{proxy}} \leftarrow \bigcup_{i\in\mathcal{C}_{\text{mal}}} \mathcal{D}_i \setminus \mathcal{D}_{\text{unlearn}}$

Steps (a)–(d) below run in parallel after stabilization, still within the training phase.

(a) Influence-based trajectory estimation via LISSA ▷ Eqs. (4)–(5)

$\mathbf{b} \leftarrow \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_{\text{unlearn}}; \mathbf{w}_{t_s})$; $\mathbf{h}_0 \leftarrow \mathbf{b}$

for $j = 1$ **to** J **do**
  sample mini-batch $z_j \subset \mathcal{D}_{\text{proxy}}$
  $\mathbf{h}_j \leftarrow \mathbf{b} + \big(I - \eta_{\text{L}}(\nabla^2 \mathcal{L}(z_j; \mathbf{w}_{t_s}) + \lambda_{\text{LISSA}} I)\big)\mathbf{h}_{j-1}$
end for
$\mathbf{v}_{\text{inf}} \leftarrow \mathbf{h}_J$

(b) Knowledge vacuity profiling ▷ Eqs. (6)–(7)

for each profiled neuron a **do**
  $$\text{Vacuity_Score}(a) \leftarrow \frac{\mathbb{E}_{\mathbf{x}\in\mathcal{D}_{\text{unlearn}}}[|a(\mathbf{x})|]}{\mathbb{E}_{\mathbf{x}\in\mathcal{D}_{\text{proxy}}}[|a(\mathbf{x})|] + \epsilon}$$
end for
for each parameter d: $\text{Vacuity_Score}_d \leftarrow \max_{a\in\mathcal{N}(d)} \text{Vacuity_Score}(a)$

(c) Saliency-guided mask generation ▷ Eqs. (8)–(9)

for each d: $s_d \leftarrow |\widehat{(\mathbf{v}_{\text{inf}})_d}| \cdot \widehat{\text{Vacuity_Score}}_d$

$M_{\text{VOID}}(d) \leftarrow \mathbb{1}[s_d \geq s_{(\lceil KP \rceil)}]$ for all d, where P is the total number of parameters

(d) Shadow teacher training ▷ Sec. 4.3.1

$\mathbf{w}_{\text{tea}} \leftarrow \mathbf{w}_{t_s}$; $\mathcal{D}_{\text{trigger}} \leftarrow \{(\text{trigger}(\mathbf{x}), y_{\text{tar}}) : \mathbf{x} \sim \mathcal{D}_{\text{proxy}}\}$

optimize $\mathcal{L}_{\text{tea}}(\mathbf{w}_{\text{tea}})$ on $\mathcal{D}_{\text{proxy}} \cup \mathcal{D}_{\text{trigger}}$ with weight γ_{tea}

request unlearning of $\mathcal{D}_{\text{unlearn}}$; let $\mathbf{w}_T$ denote the FL-converged model

Phase 2: Unlearning

Convention: $\tilde{\mathbf{w}}_r$ denotes the broadcast model received at round r, with $\tilde{\mathbf{w}}_1 = \mathbf{w}_T$. Set $\mathcal{L}_{push} = 0$ when $r = 1$ (no prior drift observable).

each malicious client $i \in \mathcal{C}_{\text{mal}}$ constructs local $\mathcal{D}^i_{\text{trigger}}$ by applying the trigger to client i's data with label y_{tar}

for $r = 1$ **to** R **do**
  receive global broadcast model $\tilde{\mathbf{w}}_r$
  for each malicious client $i \in \mathcal{C}_{\text{mal}}$ **in parallel do**
    initialize $\hat{\mathbf{g}}^i_r \leftarrow \mathbf{0}$
    for local step $= 1$ **to** E_{atk} **do**
      evaluate $\mathcal{L}_{\text{push}}$, $\mathcal{L}_{\text{pull}}$, $\mathcal{L}_{\text{unlearn}}$ at $\tilde{\mathbf{w}}_r + \hat{\mathbf{g}}^i_r$ ▷ Eqs. (10)–(12)
      $\mathcal{L}_{\text{VOID}} \leftarrow \mathcal{L}_{\text{unlearn}} + \alpha\,\mathcal{L}_{\text{push}} + \beta\,\mathcal{L}_{\text{pull}}$ ▷ Eq. (13)
      $\hat{\mathbf{g}}^i_r \leftarrow \hat{\mathbf{g}}^i_r - \eta_{\text{atk}}\, M_{\text{VOID}} \odot \nabla_{\hat{\mathbf{g}}^i_r} \mathcal{L}_{\text{VOID}}$ ▷ Eq. (14)
    end for
    submit $\hat{\mathbf{g}}^i_r$ to the server for aggregation
  end for
end for

F Dataset Description and Licenses

We conduct experiments on four public benchmark datasets spanning image, sensor, and text classification: MNIST, CIFAR-10, Human Activity Recognition (HAR), and AG News. Table 4 summarizes their key statistics, model pairing, and license or usage terms.

Table 4: Summary of datasets used in our experiments.

Dataset	Modality	Train/Test	Classes	Model	License / Terms
MNIST	Image	60,000 / 10,000	10	CNN	CC BY-SA 3.0
CIFAR-10	Image	50,000 / 10,000	10	ResNet-18	CC BY 4.0, as listed by UCI
HAR	Sensor time series	7,352 / 2,947	6	HAR-CNN	CC BY 4.0, as listed by UCI
AG News	Text	120,000 / 7,600	4	FastText	Non-commercial research use; HF wrapper license unknown

MNIST [31]. MNIST is a handwritten digit recognition dataset containing 70,000 grayscale images of digits from 0 to 9. Each image has size 28×28 with one channel, and the standard split contains 60,000 training examples and 10,000 test examples. In our experiments, MNIST is paired with a lightweight CNN. *Dataset license:* Creative Commons Attribution-Share Alike 3.0 (CC BY-SA 3.0), as listed by TensorFlow/Keras. *URL:* `https://yann.lecun.com/exdb/mnist/`. *License metadata:* `https://www.tensorflow.org/api_docs/python/tf/keras/datasets/mnist/load_data`.

CIFAR-10 [29]. CIFAR-10 is an image classification dataset consisting of 60,000 color images in 10 object classes, with 6,000 images per class. Each image has size 32×32 with three RGB channels, and the standard split contains 50,000 training examples and 10,000 test examples. We use ResNet-18 for CIFAR-10 experiments. *Dataset license:* Creative Commons Attribution 4.0 International (CC BY 4.0), as listed by the UCI Machine Learning Repository metadata for CIFAR-10. *URL:* `https://www.cs.toronto.edu/~kriz/cifar.html`. *License metadata:* `https://archive.ics.uci.edu/dataset/691/cifar+10`.

Human Activity Recognition (HAR) [3]. The UCI HAR dataset contains smartphone inertial sensor recordings from 30 volunteers performing six daily activities: walking, walking upstairs, walking downstairs, sitting, standing, and laying. Signals were captured with accelerometer and gyroscope sensors at 50Hz and segmented into fixed-width windows, yielding 10,299 labeled examples under the standard split of 7,352 training and 2,947 test examples. We pair HAR with a HAR-CNN architecture. *Dataset license:* Creative Commons Attribution 4.0 International (CC BY 4.0), as listed by the UCI Machine Learning Repository. *URL:* `https://archive.ics.uci.edu/dataset/240/human+activity+recognition+using+smartphones`.

AG News [49]. AG News is a four-class news topic classification benchmark derived from the AG corpus of news articles. The benchmark uses the four largest classes, with 30,000 training samples and 1,900 test samples per class, for a total of 120,000 training and 7,600 test examples. We use FastText for AG News experiments. *Dataset terms:* the original AG corpus is provided for research and other non-commercial activities; the commonly used HuggingFace distribution reports its dataset license as unknown. *URL:* `https://huggingface.co/datasets/fancyzhx/ag_news`. *Terms metadata:* `https://www.tensorflow.org/datasets/catalog/ag_news_subset`.

G Default Hyperparameter Settings

Table 5 lists the default hyperparameter values referenced in Section 5. Unless explicitly overridden by a particular study, every experiment uses these settings.

H Additional Experimental Studies

This appendix describes additional studies that did not fit within the main paper's space budget. All experiments follow the default configuration in Appendix G unless otherwise stated.

Runtime Overhead. Table 6 shows the cumulative client-side runtime of each attack in seconds, decomposed into offline preparation, FL training, and FU stages under the same client configuration. This metric measures the total computation performed by selected clients rather than the wall-clock

Table 5: Default hyperparameter settings used throughout the experiments.

Group	Symbol / Name	Default Value
System / FL setup		
	Default dataset	MNIST
	Default FU protocol	FedRR
	Data partition	IID
	Number of clients (n)	100
	Malicious clients (m)	$m = 20$
	Per-round client sampling rate	80%
VOID optimization		
	Optimizer	SGD
	Attack learning rate (η_{atk})	0.01
	Momentum	0.9
	Local steps per round (E_{atk})	5
	Numerical stability constant (ϵ)	10^{-8}
	Stabilization tolerance	1%
	Stabilization window	10 consecutive rounds
Mask generation		
	LISSA iterations (J)	15
	LISSA damping (λ_{LISSA})	0.01
	Mask fraction (K)	5% of total parameters
Shadow teacher and unlearning-phase objective		
	Trigger weight (γ_{tea})	1.0
	Push weight (α)	1.0
	Pull weight (β)	1.0
	Unlearning weight (μ)	0.5
	Distillation temperature (τ)	4.0

Table 6: Runtime decomposition of FU-targeted attacks (in seconds).

Attack	Offline preparation	Training phase	Unlearning phase
VOID	504	114947	63105
BadUnlearn	0	116312	58324
FUBA	0	162160	56842
CP-FU	1381	114035	57158
w/o attack	0	116854	57842

latency under parallel execution. In the offline stage, VOID spends 504s on constructing the attack mask and preparing the auxiliary teacher model, whereas CP-FU has the largest offline cost of 1381s due to camouflage-data generation. During FL training, all methods except FUBA remain close to the no-attack baseline, with only minor runtime variations. By contrast, FUBA incurs the largest training cost of 162160s due to its additional online attack procedures, including repeated auxiliary model training and adversarial update adjustment. VOID takes slightly longer in the FU stage because sparse masked injection is performed during unlearning, but its main overhead remains a one-time preparation cost.

Impact of Unlearn Data Ratio. Figures 10 and 11 show the ASR and ACC of VOID as the per-client unlearn data ratio $|\mathcal{D}^i_{\text{unlearn}}|/|\mathcal{D}_i|$ varies in $\{1\%, 2\%, 5\%, 10\%, 20\%\}$ across the three FU protocols. The attack becomes substantially stronger as the forgotten subset grows from 1% to 2%. When only 1% of each malicious client's local data is requested for unlearning, the induced vacuity is limited and protocol-dependent, with FedRR and FedAU providing only a small writable subspace and therefore visibly lower ASR, while NoT is already more vulnerable due to its more aggressive parameter negation. Increasing the ratio to 2% sharply raises ASR for all protocols, and ratios of 5% or larger drive the attack to near-perfect success. This confirms that larger unlearn data ratios amplify the low-curvature residual dimensions that VOID exploits, but also shows that only a modest forgotten subset is sufficient to create exploitable capacity. In contrast, ACC remains nearly unchanged across all unlearn data ratios, staying around 98% for FedRR, NoT, and FedAU. Thus, increasing the unlearn data ratio primarily improves backdoor writeability rather than degrading clean-task utility.

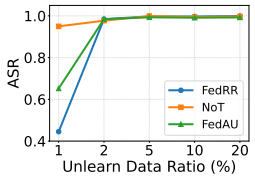

Figure 10: ASR under different unlearn data ratios.

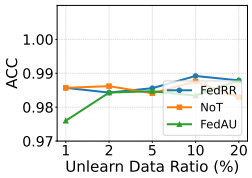

Figure 11: ACC under different unlearn data ratios.

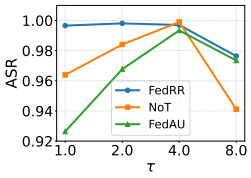

Figure 12: ASR under different distillation temperatures.

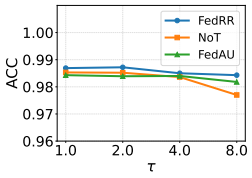

Figure 13: ACC under different distillation temperatures.

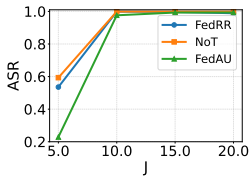

Figure 14: ASR under different LISSA depths.

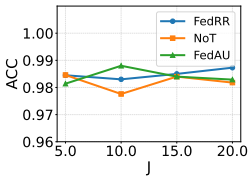

Figure 15: ACC under different LISSA depths.

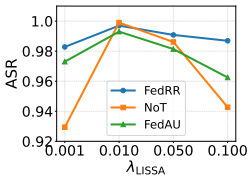

Figure 16: ASR under different LISSA damping values.

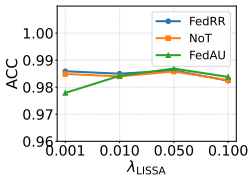

Figure 17: ACC under different LISSA damping values.

Effect of Distillation Temperature τ. Figures 12 and 13 show the impact of the shadow-teacher temperature τ. ASR exhibits a non-monotonic trend. For NoT and FedAU, increasing τ from 1 to 4 substantially strengthens the attack, while FedRR remains close to saturation across small and moderate temperatures. All three FU protocols reach near-perfect attack performance at $\tau = 4$. When τ is further increased to 8, ASR drops for all protocols, most noticeably under NoT, where ASR decreases from 0.999 at $\tau = 4$ to 0.941 at $\tau = 8$. In contrast, ACC remains stable around 98% across all τ values, with only a slight decrease at the largest temperature. These results indicate that a moderate temperature provides the best balance between soft-label guidance and trigger-driving strength.

Effect of LISSA Depth J and Damping λ_{LISSA}. Figures 14 and 15 study the number of LISSA iterations J, which controls the quality of the inverse-Hessian-vector approximation used for mask construction. With only $J = 5$ iterations, ASR is clearly lower, especially under FedAU, indicating that a shallow recursion provides an insufficiently accurate trajectory estimate and misses part of the writable subspace. Increasing J to 10 sharply improves ASR for all three FU protocols, and further increasing J to 15 or 20 yields little additional gain because the attack is already close to saturation. ACC remains stable around 98% across all J values, showing that a more accurate mask improves attack effectiveness without sacrificing clean utility. Figures 16 and 17 further evaluate the LISSA damping term λ_{LISSA}. ASR peaks around $\lambda_{\text{LISSA}} = 0.01$, where damping is sufficient to stabilize the stochastic Hessian inverse approximation while still preserving sensitivity to low-curvature forgotten-data directions. Smaller damping can make the estimate noisier, whereas larger damping over-regularizes the inverse-Hessian signal and weakens the contrast between vacuous and well-constrained dimensions. ACC changes only mildly across damping values, confirming that λ_{LISSA} mainly affects the quality of the attack mask rather than the retained-task performance.

I Limitations and Broader Impact

Limitations. Our study evaluates VOID on representative calibration-based FU protocols and standard classification benchmarks spanning image, sensor time-series, and text modalities. These settings cover several common approximate FU scenarios, and extending the analysis to other unlearning designs, larger model families, and production systems with additional auditing or access-control mechanisms is an important direction for future work.

Broader impact. This work is dual-use, but its primary contribution is to expose a previously underexplored security dimension of federated unlearning, namely that the forgetting process itself can create writable residual structure that adversaries may exploit. By identifying this mechanism and operationalizing it through VOID, our study provides a concrete stress test for future FU systems and

highlights the need for defenses that go beyond training-phase robustness assumptions. In particular, the results suggest that secure FU should audit not only whether data influence is removed, but also how unlearning-phase updates reshape low-curvature residual dimensions and whether post-unlearning models acquire unintended trigger semantics. The experiments use public benchmark datasets and controlled research settings rather than real user data.

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 abstract and introduction accurately summarize the proposed VOID attack, the knowledge vacuity motivation, and the main empirical findings. These claims are supported by the method, analysis, and experiments within the calibration-based FU setting studied in the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses its scope through the threat model and the calibration-based FU setting. Appendix I further discusses the evaluated benchmark scope and future directions beyond the studied FU protocols, model families, and deployment settings.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The theoretical results are stated as numbered propositions and derivations with explicit assumptions such as local quadratic approximation, Hessian eigendecomposition, and stable LISSA damping. The supporting proofs and derivations are provided in the appendix and cross-referenced from the main text.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The paper specifies the datasets, model architectures, FU protocols, attack settings, evaluation metrics, client configuration, training and unlearning rounds, and random seeds. Default optimization and attack hyperparameters are also listed in the appendix.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: A code package with instructions is submitted as supplementary material for review. The experiments use public benchmark datasets, and the supplement will describe the commands and configurations needed to reproduce the main results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper specifies the datasets, data partitioning, model architectures, FU protocols, attack setup, evaluation metrics, client configuration, and training and unlearning rounds. Additional optimizer and attack hyperparameters are provided in the appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

Justification: The main results are averaged over three random seeds, and the vacuity verification analysis reports 95% bootstrap confidence intervals. However, due to space constraints, the main ASR and ACC tables do not include full error bars or statistical significance tests.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper reports runtime overhead for the evaluated attacks and states that experiments use one RTX 4090 GPU with 24GB memory.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work studies a security risk in a controlled research setting using public benchmark datasets. Its purpose is to identify vulnerabilities in federated unlearning and motivate FU-specific defenses.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Appendix I discusses the dual-use nature of the work and its positive impact in exposing a previously underexplored FU security dimension that can guide future defenses.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release high-risk models, scraped datasets, or other assets requiring controlled access safeguards. The experiments use public benchmark datasets.

    Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: The paper cites the public benchmark datasets and model architectures used in the experiments. Appendix F explicitly summarizes the datasets, URLs, and license or usage terms, including the non-commercial research-use terms and unknown wrapper license for AG News.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: The paper introduces code for reproducing the proposed method and experiments. An anonymized supplementary package will document the implementation, configurations, and reproduction instructions.

Guidelines:

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

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

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

Answer: [N/A]

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

Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were used only for writing, editing, or formatting assistance and were not part of the core method, experiments, or scientific claims.

    Guidelines:

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