ACTIVE UNLEARNING PROBING: LABEL INFERENCE ATTACK FROM THE CLIENT'S VANTAGE IN FEDERATED UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated unlearning allows a client to request that its data be retracted from a jointly trained model, after which the server broadcasts the updated model to all participants. Prior forgotten-label inference typically assumes an attacker with a server vantage that can read individual client updates, yet in practice the server is usually trusted, so we ask whether the same inference attack can be mounted from the vantage of an ordinary client. Such a threat model is less restrictive and easier to realize. We show that the answer is yes. We propose **Active Unlearning Probing (AUP)**, which issues short unlearning probes as a legitimate client to build a per-class gradient template, then matches the classifier-layer direction of the observed global-model change against these templates to infer which label class another user has forgotten. Using only its own labeled training data, with no shadow set, no labeled test set, and no queries to the victim, AUP identifies the forgotten label at 0.96 accuracy on CIFAR-10 and 0.80 on MNIST, likewise identifies labels on CIFAR-100 with a ViT-B/16 and on the Yahoo Answers text dataset, recovers three simultaneously deleted classes at an intersection-over-union (IoU) of 0.58, and remains effective under two different unlearning mechanisms. We also find that AUP becomes more effective as the federation grows, a phase transition governed by the signal-to-noise ratio rising with the number of clients. Code is available at `https://anonymous.4open.science/r/AUP`.

1 INTRODUCTION

Federated learning (FL) trains a shared model across multiple clients that exchange only model updates and never raw data, with aggregation of the global model serving as the privacy boundary (McMahan et al., 2017; Kairouz & McMahan, 2021). In deployments such as Gboard and Apple keyboard prediction, cross-hospital consortia, and inter-bank fraud detection (Bonawitz et al., 2019; Xu et al., 2021), no participating client can observe any other client's raw data, gradients, or local updates. Regulations like GDPR and CCPA (Protection, 2018; Harding et al., 2019) grant clients a right to be forgotten, so FL systems increasingly support federated unlearning (FU) (Liu et al., 2021; Wu et al., 2022; Shaik et al., 2024), in which a client issues a forget request that steers the model to eliminate the influence of the target data while training continues.

Recent work shows that unlearning itself carries a leakage risk. The perturbation a forget request leaves on the model is class-correlated, so a curious observer can recover the label a client has asked to erase. These attacks, however, are posed from the server's vantage, a semi-honest aggregator that stores each client's update Δ_{local} across rounds and isolates the victim's contribution by subtracting it from the aggregate. ULIA (Wang et al., 2025a) and FUIA (Zhou et al., 2026) recover the forgotten label from Δ_{local} this way. Both require visibility into Δ_{local}, and both therefore presume privileged infrastructure access. In production, by contrast, the aggregator is generally a trusted, audited component, while the attacker is more often an ordinary participant, a legitimately registered client that sees only the aggregated global model w^t each round and never any peer's update. Whether such an attacker can infer the labels other clients forget from this client-side vantage is unstudied, and it faces three challenges.

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

- **C1.** Server-vantage attacks isolate the signal by subtracting the victim's uploaded update Δ_{local}, which a client can never observe.
- **C2.** A peer client has no labeled data for shadow training or calibration and cannot query the victim to obtain any.
- **C3.** Client subsampling, momentum, and learning-rate decay inject optimization noise into the aggregated update, burying any single client's contribution.

To address these challenges, we propose Active Unlearning Probing (AUP), a forgetting label inference attack initiated entirely from the client. Since ordinary clients may issue forget requests, an attacker can build its own reference signals. To address C1, AUP runs a short unlearning campaign on each candidate class and records the change in the broadcast global model, replacing the per-client local update it can never observe with the aggregate-level perturbation it legitimately receives. To address C2, AUP averages these self-induced changes into a per-class template and identifies the victim's forgotten class by matching the classifier-layer direction of the observed global-model change against the templates via cosine similarity, requiring no shadow training and using only its own data. To address C3, restricting observation to the classifier layer concentrates the class-relevant signal and suppresses the system noise that would otherwise mask it.

We evaluate AUP over a range of settings spanning architectures and modalities, including three image datasets, MNIST (LeCun et al., 1998) on a CNN, CIFAR-10 (Krizhevsky et al., 2009) on a ResNet-18 (He et al., 2016), and CIFAR-100 on a ViT-B/16 transformer (Dosovitskiy et al., 2020), together with the text dataset Yahoo Answers (Zhang et al., 2015), covering two FU mechanisms, gradient ascent (Halimi et al., 2022) and SGA-EWC (Wu et al., 2022). As an ordinary client that observes only the aggregated broadcast and holds no labeled test data, AUP raises forgotten-class identification accuracy to 0.96 on CIFAR-10 and 0.80 on MNIST, and reaches IoU 0.58 in the multi-class (L=3) forgetting setting. It likewise reaches 0.96 in the 100-class ViT/CIFAR-100 setting and 1.00 in the text setting on Yahoo Answers, well above the random-guess floor. AUP by design targets unlearning mechanisms that imprint a class-specific direction on the shared model. Once Δ_{local} is hidden, ULIA's subtraction step loses its operand and FUIA loses the per-client signal its inversion relies on, so neither prior attack transfers to the ordinary-client vantage that AUP occupies.

We further identify a federation-size scaling phenomenon. On CIFAR-10/ResNet-18 the attack undergoes a phase transition as the signal-to-noise ratio (SNR) crosses an order-one threshold, and strengthens as the federation grows. Prior work indicates that passive leakage through the aggregate decays as $O(1/N)$ in the number of users (Elkordy et al., 2022; Zhao et al., 2024). Our experiments show instead a per-block SNR proxy that rises rather than falls with the federation, tracking this strengthening. De-confounding controls rule out sampling, data-volume, and model-quality confounds, and an SNR identity marks the threshold and collapses the data onto a common curve, so larger federations are the more exposed ones. The same regularity reproduces on transformers across modalities, including a ViT-B/16 at 100-class scale (CIFAR-100, $100\times$ the random floor) and a text encoder (MiniLM on Yahoo Answers), both reaching 1.00. Our contributions are as follows.

- We formalize forgotten-label inference under an ordinary-client vantage that is strictly weaker than the server vantage, and propose AUP, a classifier-layer forgotten-label inference attack that requires no shadow training and uses only the attacker's own data.
- We conduct numerous experiments and AUP successfully recovers unlearning class labels across two CNN/ResNet image datasets, a 100-class ViT/CIFAR-100 setting, a text dataset, and two gradient-based FU mechanisms.
- We report a federation-size scaling transition, found experimentally, under which larger federations are easier to attack, with the onset at an order-one SNR. We support it with de-confounding controls, including a model-quality check, and an SNR identity, and show that it reproduces across ResNet, a ViT transformer at 100 classes, and a text encoder.

2 RELATED WORK

Machine unlearning removes the influence of specific training data from a trained model without a full retrain, through exact or approximate schemes (Cao & Yang, 2015; Bourtoule et al., 2021; Ginart et al., 2019; Golatkar et al., 2020; Graves et al., 2021; Nguyen et al., 2025), and federated unlearning carries

this into the FedAvg setting, where a client asks the server to erase its contribution from the shared model while training continues (Liu et al., 2022; Wang et al., 2022; Che et al., 2023; Gao et al., 2024). Most attacks that recover the forgotten label operate from the server side, where the individual client updates are visible before aggregation. ULIA (Wang et al., 2025a) subtracts the victim's own update Δ_{local} from the round aggregate and maps the isolated residual to a class, while FUIA (Zhou et al., 2026) scores the forgotten class from the output-layer weights and bias and then reconstructs the erased samples by gradient inversion. Both depend on per-round access to Δ_{local}, or equivalently to the global model immediately before and after the unlearning step, and therefore assume an adversary with privileged infrastructure access rather than an ordinary participant; white-box membership inference on the same pre/post-unlearning gap (Nasr et al., 2019) sits in the same regime.

Table 1: FU-attack capabilities. *Ord*, ordinary benign client (not a poisoner); *No*-Δ_{local}, no access to any victim update; *Free*, no shadow or test data; *Hon*, only honest protocol requests.

Method	Ord	No-Δ_{local}	Free	Hon
ULIA	✗	✗	✓	✓
FUIA	✗	✗	✓	✓
Retaliatory	✗	✓	✓	✗
Verifiable Forgotten	✗	✗	✗	✓
White-box MIA	✗	✗	✗	✓
FedMUA	✗	✓	✓	✗
Trail Erasure	✗	✓	✓	✗
BadUnlearn	✗	✓	✓	✗
BadFU	✗	✓	✓	✗
FedCPA	✗	✓	✓	✗
AUP (Ours)	✓	✓	✓	✓

Another group of attacks acts as a client and corrupts the model through the forget interface, attacking integrity rather than privacy. FedMUA (Chen et al., 2025) forces targeted misclassifications through maliciously crafted forget requests, and poisoning-through-unlearning attacks, Hidden Poison (Di et al., 2023), its federated form FedCPA (Lai et al., 2026), BadFU (Lu et al., 2025), BadUnlearn (Wang et al., 2025b), and Trail Erasure (Alam et al., 2024), plant backdoors that a deletion request later triggers or conceals. Retaliatory (Sheng et al., 2026) shares our client-side, Δ_{local}-free vantage but remains a poisoning attack that uploads updates after a detection phase to reconstruct data. AUP keeps that vantage while staying honest and targeting label privacy rather than integrity.

Beyond federated unlearning, membership (Shokri et al., 2017; Carlini et al., 2022), property (Melis et al., 2019), and gradient-inversion (Zhu et al., 2019; Geiping et al., 2020) attacks on FL need shadow models, labeled data, or the shared gradients; model inversion recovers inputs from confidence (Fredrikson et al., 2015), and source inference (Athanasiou et al., 2026) identifies a point's owner from individual updates. Unlearning also leaves traces in the centralized pre/post gap (Ma et al., 2026), an LLM's outputs (Chen et al., 2026), or image generators (Liu et al., 2026b), and vertical-FL label unlearning (Gu et al., 2026), verifiable federated unlearning (Zhang et al., 2025), and robust image-generation unlearning (Liu et al., 2026a) are studied defensively. Forgetting is thus information-bearing, yet all of these assume server-side access or a centralized model, the gap that recent FU-privacy surveys identify (Jeong et al., 2026; Wang et al., 2023) and that leaves no client-vantage baseline to compare against. Table 1 compares these lines on four capability axes, where AUP is the first to satisfy all four.

3 PRELIMINARIES

Federated learning. Federated learning (FL) (McMahan et al., 2017; Konečnỳ et al., 2016; Li et al., 2020; Reddi et al., 2020) trains a shared model w across C clients that keep their data local, minimizing the data-weighted objective

$$F(w) = \sum_i \frac{|D_i|}{|D|} F_i(w), \qquad F_i(w) = \frac{1}{|D_i|} \sum_{(x,y)\in D_i} \ell(w; x, y). \tag{1}$$

FedAvg proceeds in rounds. The server broadcasts w^t to a sampled subset S^t, each client $i \in S^t$ returns a local update $\Delta_i^t = w_i^{t+1} - w^t$, which is that client's individual contribution and which we write Δ_{local}, and the server aggregates

$$w^{t+1} = w^t + \sum_{i\in S^t} \frac{|D_i|}{\sum_{j\in S^t} |D_j|} \Delta_i^t. \tag{2}$$

A client therefore observes only the aggregated broadcast w^t, and never any peer's Δ_{local} (Bonawitz et al., 2017).

Federated unlearning. Federated unlearning (FU) removes the influence of selected data from a trained model, realizing the right to be forgotten without a full retrain. Given a model trained on $\bigcup_i D_i$ and a forget set $D^{\mathrm{f}} \subseteq D_k$, a mechanism $\mathcal{U}$ returns

$$w^{\mathrm{u}} = \mathcal{U}(w; D^{\mathrm{f}}) \;\approx\; \arg\min_{w} \sum_{(x,y)\in\bigcup_i D_i \setminus D^{\mathrm{f}}} \ell(w; x, y), \tag{3}$$

which approximates the model obtained by retraining on the retained data alone. We study **class-level** unlearning, where $D^{\mathrm{f}} = \{(x, y) \in D_k : y \in Y_{\mathrm{forget}}\}$, and consider two mechanisms. GA (Halimi et al., 2022) ascends the loss on D^{f}

$$w \leftarrow w + \eta \nabla_w \mathcal{L}(w; D^{\mathrm{f}}). \tag{4}$$

SGA-EWC (Wu et al., 2022) adds penalty $\lambda \sum_p F_p \, (w_p - w_p^\star)^2$ (Kirkpatrick et al., 2017) that protects the retained classes. AUP applies to mechanisms like these that leave a class-specific direction on the aggregate; retraining-based erasure that descends only on the retained data, such as FedEraser (Liu et al., 2021), lies outside its scope.

Threat model. We consider a federation comprising a trusted aggregator, a victim B, and an attacker A that is a legitimately enrolled client. A follows FedAvg and the FU API strictly, never issuing malformed updates, never poisoning, and never deviating from the protocol, but it *actively* probes each candidate class, either on its own saved copy of the broadcast or through legitimate `forget_class(c)` requests, alongside its benign training updates. This adversary is stronger than a purely passive honest-but-curious server-side observer, yet far less destructive than a Byzantine client (Blanchard et al., 2017), and in its deployable form it issues no real forget requests, so it perturbs no shared model and the server sees only ordinary participation. A receives each round's aggregated model w^t, identical to what every honest client receives, and has access to no peer's update, no server state, and no test set. Reading only this aggregate, AUP is untouched by Secure Aggregation (Bonawitz et al., 2017), which by hiding per-client updates instead disables the server-vantage attacks. It uses only its own labeled training data, held as an ordinary enrolled client, with no shadow dataset and no labeled test set. The threat model further rests on two premises: **(i)** The unlearning direction of a given class is essentially consistent across clients, which we realize through an IID partition with an identical label set on every client. Non-IID heterogeneity, Byzantine aggregators, and heterogeneous label sets are outside the scope of this paper, and extending the attack to non-IID federations is our primary future direction (Section 6). **(ii)** The victim B conducts its forgetting within its own unlearning window as a continuous, self-initiated multi-round campaign rather than a passively subsampled training round, so its per-round contribution accumulates coherently throughout that period. This continuous-participation assumption keeps the victim's signal coherent while honest updates remain subsampled, so the federation-size effect of Section 5.4 is scoped to such sustained forget campaigns.

Problem formulation. Given only its broadcast transcript $\{w^t\}_t$, and with no access to Δ_{local}, the attacker A must output a set $\hat{Y} \subseteq \mathcal{Y}$ with $|\hat{Y}|{=}L$ that matches the victim's forgotten set Y_{forget}. We report exact-match accuracy for the single-class case and Intersection-over-Union for multi-classes,

$$\mathrm{Acc} = \mathbf{1}\left[\hat{Y} = Y_{\mathrm{forget}}\right], \qquad \mathrm{IoU} = \frac{|\hat{Y} \cap Y_{\mathrm{forget}}|}{|\hat{Y} \cup Y_{\mathrm{forget}}|}. \tag{5}$$

The client vantage is informationally weaker than the server's, since every client decoder can be simulated by the server, so that $\mathrm{acc}^\star_{\mathrm{server}} \geq \mathrm{acc}^\star_{\mathrm{client}}$ holds unconditionally. It is nonetheless the more plausible vantage in deployment, because production aggregators operate under service-level agreements, auditing, and attestation (Bonawitz et al., 2019; Xu et al., 2021), which leaves the peer client as the residual threat surface.

4 METHODOLOGY

4.1 OVERVIEW

We propose AUP. A client attacker cannot see the victim's update but can issue forget requests of its own. It unlearns each candidate class from its own data, synthesizes a per-class reference (Snell

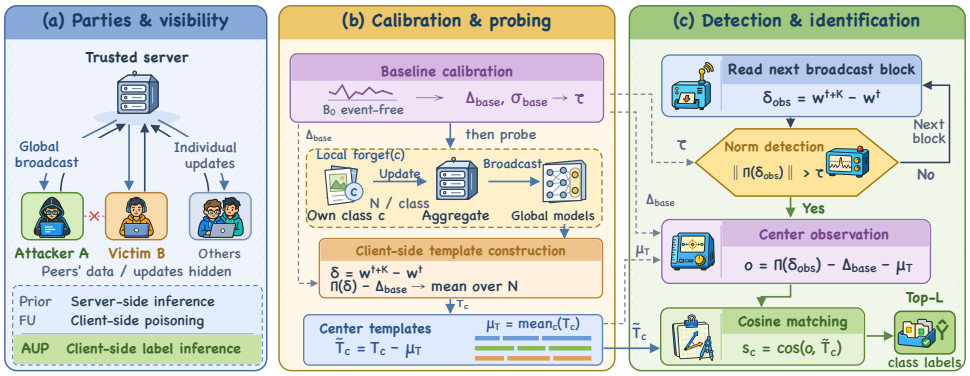

Figure 1: Overview of AUP. (a) Parties and the client attacker's visibility. (b) Baseline calibration and active probing to construct baseline-centered class templates. (c) Event detection and cosine matching to infer the top-L forgotten classes; monitoring continues if no event is detected.

Algorithm 1 Active Unlearning Probing (AUP)

Require: candidate classes $\mathcal{Y}$; probes per class N; block length K; baseline blocks B_0; detection constant k; forgotten count L

record B_0 event-free blocks; $\Delta_{\text{baseline}} \leftarrow$ mean of $\Pi(\delta^b)$, $\tau \leftarrow \text{mean}(\|\Pi(\delta^b)\|) + k\,\sigma_{\text{base}}$ ▷ calibrate
for each class $c \in \mathcal{Y}$ **do**
  run N short forget campaigns on own class-c data
  $T_c \leftarrow \frac{1}{N}\sum_{i=1}^{N}\left(\Pi(\delta^{(c,i)}) - \Delta_{\text{baseline}}\right)$ ▷ per-class template
end for
for each monitored block with delta δ_{obs} **do**
  if $\|\Pi(\delta_{\text{obs}})\| > \tau$ **then** ▷ event detected
    $o \leftarrow \Pi(\delta_{\text{obs}}) - \Delta_{\text{baseline}}$
    return $\hat{Y} = \arg\text{top-}L_{c\in\mathcal{Y}}\ \cos(o, T_c)$
  end if
end for

et al., 2017), and matches the victim's unlearning-induced change in the broadcast model against these references. AUP observes only the classifier layer $\Pi = \{\texttt{fc.weight}, \texttt{fc.bias}\}$, where the per-class signal concentrates and where the representation-level noise of the backbone averages out under aggregation. The attacker reads the federation in blocks of K consecutive rounds and summarizes each block by its cumulative change $\delta = w^{t+K} - w^t$, of which AUP attends only to the classifier-layer part $\Pi(\delta)$, and every AUP operation is defined on this classifier-layer block delta. Figure 1 illustrates the framework, and Algorithm 1 gives its probe, calibrate, and monitor procedures. AUP costs $|\mathcal{Y}| \times N$ unlearning probe campaigns to build the templates, followed by $O(T_{\text{blocks}}\,|\mathcal{Y}|)$ cosine evaluations during monitoring. Per-phase pseudocode is in Appendix A.

4.2 Active Probing and Template Construction

For softmax cross-entropy $\mathcal{L} = -\log p_y$ with $p = \text{softmax}(W\phi(x) + b)$ and penultimate features $\phi(x)$, the gradient on the classifier row W_c is conditioned on the label (Li et al., 2021; Fu et al., 2022),

$$\frac{\partial \mathcal{L}}{\partial W_c} = \left(p_c - \mathbf{1}[y{=}c]\right)\phi(x)^\top. \tag{6}$$

Unlearning class c negates this term and leaves a coherent direction on the row W_c, whereas the backbone carries representation-level information that averages out across classes after aggregation. The decisive property is transferability. Once averaged over class-c samples, the direction of (6) depends only on the per-class mean feature $\mathbb{E}[\phi(x) \mid y{=}c]$ and on the shared broadcast model, and not on which client holds the data. Under premise (i) of Section 3, the attacker's probe on class c therefore reproduces the signature that the victim's own unlearning of c leaves in the aggregate.

Over the first B_0 blocks of its enrollment window, which it assumes free of forget events as a change detector's warm-up null, the attacker estimates the ambient classifier-layer drift $\Delta_{\text{baseline}} = \frac{1}{B_0}\sum_b \Pi(\delta^b)$ together with the scale $\sigma_{\text{base}} = \text{std}\big(\{\|\Pi(\delta^b)\|\}_{b=1}^{B_0}\big)$, and sets the event threshold $\tau = \text{mean}\big(\{\|\Pi(\delta^b)\|\}_{b=1}^{B_0}\big) + k\,\sigma_{\text{base}}$ with k=3. It then probes. For each candidate class c it runs a short forget campaign on its own class-c data, records the resulting block delta $\delta^{(c,i)}$, and averages N probes into a baseline-centered template,

$$T_c = \frac{1}{N}\sum_{i=1}^{N}\big(\Pi(\delta^{(c,i)}) - \Delta_{\text{baseline}}\big). \tag{7}$$

The baseline subtraction removes the ambient classifier-layer drift shared by every block, leaving the per-class signature that the matcher scores in (8).

4.3 Detection and identification

During monitoring the attacker screens each block by event magnitude and flags the block when $\|\Pi(\delta_{\text{obs}})\| > \tau$. For a flagged block it centers the observation in the same way and identifies the forgotten class by cosine similarity against the templates,

$$o = \Pi(\delta_{\text{obs}}) - \Delta_{\text{baseline}}, \qquad \hat{Y} = \arg\operatorname*{top}_{c\in\mathcal{Y}}\text{-}L\ \cos\big(o, T_c\big). \tag{8}$$

We match by direction rather than by distance, because the strength of the forgetting signal varies with the victim's unknown ascent rate, campaign length, and number of forgotten samples, whereas the per-class direction is invariant and the cosine cancels this nuisance factor. Detection instead uses magnitude only to flag that an event is present, against the attacker's own noise floor, a role its unknown scale only eases. When several classes are unlearned at once, their weakly correlated signatures superimpose in δ_{obs}, with near-equiangular templates (mean off-diagonal cosine ≈ -0.1 in our classifiers, as expected under neural collapse (Papyan et al., 2020)), and the top-L scores separate them. We report identification accuracy conditional on detection, and running the same matcher on the victim's own update Δ_{local} rather than on the aggregate yields the *oracle* upper bound.

5 Experiments

5.1 Setup

Datasets. We evaluate across modalities on three image datasets and one text dataset. MNIST (LeCun et al., 1998) and CIFAR-10 (Krizhevsky et al., 2009) have ten classes each, CIFAR-100 one hundred, and Yahoo Answers (Zhang et al., 2015) is a ten-way topic task.

Backbones. A CNN on MNIST, a ResNet-18 (He et al., 2016) with GroupNorm (Wu & He, 2018) on CIFAR-10, a ViT-B/16 (Dosovitskiy et al., 2020; Deng et al., 2009) on CIFAR-100, and a MiniLM encoder (Wang et al., 2020; Reimers & Gurevych, 2019) on Yahoo Answers, all trained with FedAvg (McMahan et al., 2017) under two unlearning mechanisms, GA (Halimi et al., 2022) and SGA-EWC (Wu et al., 2022).

Baselines. No prior attack works from the client vantage, so we compare AUP against three references, the Δ_{local} oracle reading the victim's update (upper bound), the Trivial-N guesser ranking classes by their post-forget accuracy drop on N labeled test points per class, here the full test set, and the random guess at $1/|\mathcal{Y}|$ (lower bound). Trivial-N stands in for passive performance-degradation baselines, since a post-forget accuracy or loss change reads the same signal on a test set or the client's own data. AUP issues N=3 probes per class (N=5 on the CNN) over a K=5-round window.

Evaluation metrics. For single-class forgetting we report exact-match accuracy, the fraction of events whose top-scored candidate is the forgotten class, against a random floor $1/|\mathcal{Y}|$ from 0.10 to 0.01. For multi-class forgetting we report Intersection-over-Union $|\hat{Y} \cap Y_{\text{forget}}|/|\hat{Y} \cup Y_{\text{forget}}|$ between the predicted and true forgotten sets, against the random L-class floor (0.20 for L=3). Each seed evaluates five independent forget events, giving 25 events per configuration over the five seeds, and every reported number is a five-seed mean with population standard deviation.

Hardware. Eight NVIDIA A40 GPUs, two AMD EPYC 7543 processors, and 1 TB of memory, implemented in PyTorch (Paszke et al., 2019).

Table 2: Forgotten-label accuracy against the Δ_{local} oracle, the Trivial-N floor, and the random guess, mean $\pm$ std over five seeds at C=50. Upper block single-class, lower block multi-class L=3 (IoU). AUP (local) is the deployable headline; AUP (replay) is a controlled measurement.

Setting	Backbone	Metric	Random	Trivial-N	Oracle	AUP (replay)	AUP (local)
CIFAR-10 (image)	ResNet-18	Acc	0.10	$0.64 \pm .08$	$1.00 \pm .00$	$1.00 \pm .00$	$\mathbf{0.96 \pm .08}$
MNIST (image)	CNN	Acc	0.10	$0.52 \pm .16$	$1.00 \pm .00$	$0.56 \pm .20$	$\mathbf{0.80 \pm .18}$
CIFAR-100 (image)	ViT-B/16	Acc	0.01	$0.20 \pm .13$	$1.00 \pm .00$	$0.92 \pm .10$	$\mathbf{0.96 \pm .08}$
Yahoo Answers (text)	MiniLM	Acc	0.10	$0.36 \pm .20$	$1.00 \pm .00$	$1.00 \pm .00$	$\mathbf{1.00 \pm .00}$
CIFAR-10, L=3 (image)	ResNet-18	IoU	0.20	$0.41 \pm .07$	$0.66 \pm .06$	$0.53 \pm .05$	$\mathbf{0.58 \pm .05}$

5.2 Main Results

Table 2 reports the oracle, Trivial-N, and random references (Section 5.1) at C=50, where AUP maintains the ordering oracle $\geq AUP \geq$ Trivial-N $\geq$ random. We report the deployable local variant as the headline (prose below and Figure 2); AUP (replay) restarts each probe from the pretrained w_0, a controlled measurement supporting the scaling, ablation, and robustness analyses (Appendix B).

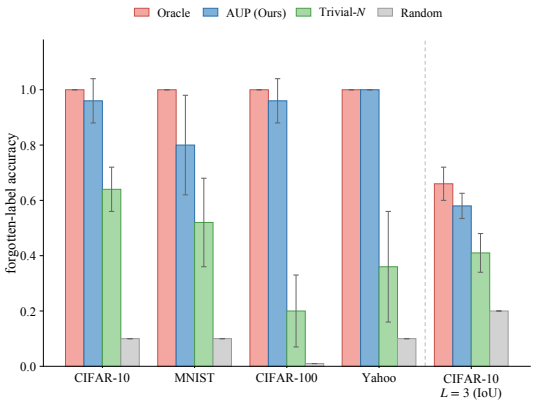

Figure 2: Baseline comparison. All AUP bars are the deployable local variant. Five-seed means, one-sd error bars.

On Yahoo Answers the local AUP matches the oracle at 1.00 over the 25 events per configuration (95% Wilson lower bound 0.87), recovering the forgotten label as reliably from the shared broadcast as an adversary that holds Δ_{local}, a 177.78% gain over Trivial-N and 900.00% over the random floor. On CIFAR-10 and the 100-class CIFAR-100 it reaches 0.96, within one standard deviation of the oracle, a 50.00% and 380.00% gain over Trivial-N and, on CIFAR-100, 96$\times$ the 0.01 random floor. MNIST is the relative weak point at 0.80, where the CNN gives classifier rows with poor separability and a faint forgetting signature, still 53.85% over Trivial-N and 700.00% over random. Trivial-N relies on a fully labeled test set, whereas AUP uses only its own training data, and at C=50 it matches or surpasses Trivial-N in every setting. For the multi-class (L=3) task the local AUP reaches IoU 0.58, a 41.46% gain over the 0.41 Trivial-N and 190.00% over the 0.20 random reference. The controlled replay measurement caps the oracle at 0.66 and its IoU rises from 0.53 at C=50 to 0.67 at C=100 (Appendix B). The gap between local AUP and the oracle is the empirical price of the client vantage, what reading the aggregate rather than Δ_{local} costs (Proposition 3). It vanishes on Yahoo Answers and stays within one standard deviation on CIFAR-10 and CIFAR-100, so on three of the four settings the broadcast leaks the forgotten label almost as freely as the victim's private update.

5.3 Generalization

Figure 3 plots AUP's match score across all candidate classes, where the forgotten class (blue) scores far above the others. Full per-seed results for every setting are in Appendix B. AUP performs well because the strength of the signature depends on how separable the classes are in the backbone's feature space. On CIFAR-10, on CIFAR-100 with a ViT, and on the topic-labeled Yahoo Answers, the per-class direction vectors are pronounced and well separated, so the cosine match saturates near the ceiling (1.00 on Yahoo Answers, 0.96 on CIFAR-10 and the hundred-class CIFAR-100).

5.4 Ablation Study

The three studies below (classifier-layer ablation, robustness, and federation-size scaling) use the AUP (replay) variant, which restarts each probe from w_0 for clean per-class deltas and follows the

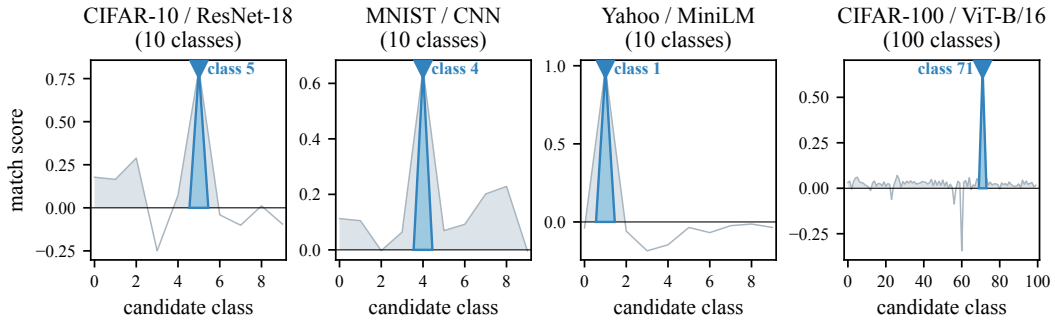

Figure 3: Per-class match score of AUP for one forget event per setting. The forgotten class (blue) scores far above every other candidate template, even among CIFAR-100's one hundred classes.

replay column of Table 2. Its CIFAR-10 accuracy at C=50 is therefore 1.00, not the local headline 0.96, and the two variants track each other across all settings (Appendix B).

Framework Ablation. We ablate the design choice at the core of AUP, its restriction of the observation to the classifier layer (Table 3). At and above the federation-size transition, the restriction is decisive. Reading the entire model instead of the classifier layer collapses exact match from 1.00 to 0.24 at C=50 and from 0.96 to 0.20 at C=100, relative drops of 76.00% and 79.17%. Below the transition, at C=10, both readers sit in the sub-transition noise near the chance floor and the ordering carries no signal. The forgetting signature is concentrated in the classifier head, under 0.05% of the ResNet-18 parameters, and diluting this tiny slice across the millions of backbone coordinates destroys it, so the whole-model reader that sees strictly more of the update does strictly worse. The restriction is also free to apply, since a client already receives the full broadcast each round and need only read its `fc.weight` and `fc.bias` coordinates while ignoring the backbone. It is thus the load-bearing design choice behind AUP rather than a tunable hyperparameter, and it is what lets the attack survive the aggregation noise of large federations.

Table 3: Ablation of the classifier-layer restriction (CIFAR-10, ResNet-18, mean ± std over five seeds, federation-replay measurement). The ablation model reads the whole network. The last column is AUP's relative gain over it.

C	Ablation model	AUP	Improvement
50	0.24 ± .08	**1.00 ± .00**	↑ 316.67%
100	0.20 ± .18	**0.96 ± .08**	↑ 380.00%

Robustness and sensitivity. We vary the remaining parameters on CIFAR-10 (Table 4). Under the two unlearning mechanisms, GA and SGA-EWC, AUP recovers the forgotten class equally well, the two differing by at most two of the 25 events across all federation sizes, so a defender cannot evade it by switching between gradient-ascent-based mechanisms. Likewise, as clients per round grows from 10 to 40 and the pretraining budget from 150 to 250, AUP declines only slightly to 0.96, the SNR staying above the identifiability threshold. No perturbation we apply drops accuracy below 0.96, within one of the 25 events of the ceiling, and since the mechanism, the cohort size, and the training budget are exactly the levers a defender controls, none of them offers an evasion. The local deployment variant matches these results on CIFAR-10 (0.96 ± .08 and 0.92 ± .10 at C=50 and C=100, all datasets in Appendix B).

Table 4: Robustness of AUP as the unlearning mechanism, clients-per-round, and training budget vary (mean ± std over five seeds, federation-replay measurement).

Factor varied	Value	AUP
Unlearning mechanism	GA	**1.00 ± .00**
	SGA-EWC	**1.00 ± .00**
Clients per round	10	**1.00 ± .00**
	25	**1.00 ± .00**
	40	0.96 ± .08
Pretraining budget	150 rounds	**1.00 ± .00**
	250 rounds	0.96 ± .08

Federation-size scaling. AUP label inference becomes easier as the federation grows, climbing from near chance at small C toward the ceiling at large C on all but MNIST (Figure 4), with

the knee shifting by difficulty, latest on the 100-class ViT ($C \approx 50$) and earliest on the text setting ($C \approx 30$). MNIST is the weakest and noisiest setting, so its realized signal does not climb smoothly: the SNR proxy `align.obs` and the accuracy both dip at C=100 and recover by C=200 (Appendix B). It stays above chance throughout (0.16 to 0.64), so the non-monotone MNIST curve reflects a weak, bumpy aggregate signal in this setting rather than a decoder failing to use it. Our experiments show this is a robust effect of federation size, not a hyperparameter artifact. Its realized driver is the per-block signal-to-noise ratio SNR (Appendix C, (9)), the victim's fixed-magnitude update set against the fluctuation scale σ_{base} of a no-event block. As C grows the accuracy rises. With clients-per-round fixed, $\alpha = \|\Delta_{\text{local}}\|/\text{cpr}$ is C-independent, so the gain comes from the denominator, cross-round cancellation among the larger honest cohort shrinking σ_{base}, which we report as an experimental finding. The global model's pre-forget accuracy is flat or decreasing in C on every dataset (Figure 5), opposite to the attack's trend. The ViT on CIFAR-100 and MiniLM on Yahoo inherit their feature separability from pretraining, yet the same transition appears on the from-scratch CNN and ResNet-18, so it is a property of the aggregation rather than of pretrained features. The transition survives when each client's data is fixed to a 500-sample shard, where AUP reaches 0.16, 0.16, 0.92, and 0.92 at C=10, 30, 50, and 100. With the cohort-size and budget controls of Table 4, it tracks SNR. The leading-order IID analysis (Appendix C) explains why a larger SNR yields identification, through template alignment, a margin transition past an $O(1)$ threshold, and server domination (Propositions 1, 2, and 3).

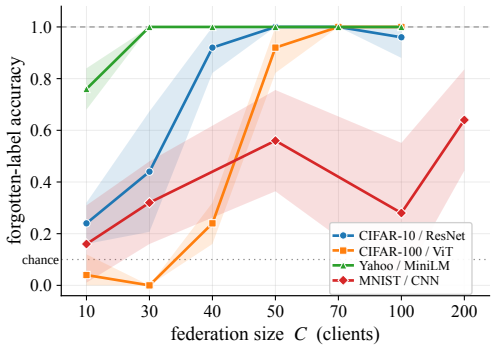

Figure 4: Federation-size scaling across datasets and modalities. Dashed line is the ceiling, dotted line the random floor, and the band is one standard deviation over five seeds.

6 FUTURE WORK

We identify three directions for future work. **(A) Data heterogeneity.** Real federations are often not IID, and per-client class skew degrades both AUP's active probing signal and the fidelity of its per-class templates. A preliminary probe on CIFAR-10 with a Dirichlet partition (Hsu et al., 2019) shows that under moderate skew (α=0.5) the non-IID accuracy at C=50 drops to 0.40, which is still well above the chance floor of 0.10. Part of the signal therefore survives, and building templates that account for skew is the priority for the next step. **(B) From labels to inputs.** AUP recovers the forgotten label but not the forgotten samples themselves. The next step is a client-vantage reconstruction attack that inverts the same classifier-layer signal into the erased inputs. **(C) Defenses.** A defender can perturb the aggregate that clients receive, for example by adding noise to the broadcast classifier layer, and preliminary tests indicate that this substantially degrades AUP. AUP could therefore be made more robust to such a perturbation by averaging over more probes, by denoising the observed delta, or by adapting the templates to a noisy channel, while also quantifying the unlearning-utility cost that any such defense must pay.

7 CONCLUSION

We have presented AUP, an active probing attack that issues forget requests from the ordinary-client vantage to build per-class templates and identify the victim's unlearned class by matching the classifier-layer direction of the observed global-model change. It requires no shadow training and uses only the attacker's own data, a threat model that assumes far less than the server vantage of prior work. Across four datasets, four backbones, and two unlearning mechanisms, AUP recovers the forgotten label well above chance and, on all but the least-separable setting, close to the Δ_{local} oracle that upper-bounds any client decoder, reaching near-perfect identification in several settings. We further report and explain a federation-size scaling effect under which AUP grows stronger as the federation grows. We hope this provides the community with a reference point for strengthening per-client unlearning privacy.

AI USE STATEMENT

Large language models were used only for language editing, including grammar correction and stylistic polishing of the manuscript. They were not used for research ideation, experimental design, data analysis, or the generation of any scientific content, and all technical contributions are the authors' own.

ETHICS STATEMENT

This work studies a privacy vulnerability in federated unlearning and follows the Code of Ethics. All experiments use publicly available benchmarks (MNIST, CIFAR-10, CIFAR-100, and Yahoo Answers) that contain no personally identifiable information and no human-subject data, so no IRB approval is required, and every attack is run against a simulated federation under our own control rather than any deployed system or real users. AUP is an offensive method, but we present it to expose a leakage channel the field has overlooked, that an ordinary client can recover the labels its peers ask to forget from the aggregated broadcast alone, and to motivate defenses for per-client unlearning privacy rather than to enable misuse.

REPRODUCIBILITY STATEMENT

All datasets are public benchmarks (MNIST, CIFAR-10, CIFAR-100, and Yahoo Answers). The assumptions, propositions, and complete proofs are in Appendix C; the AUP procedure with its oracle and local variants is in Appendix A. The experimental setup is in Section 5.1, with full hyperparameters in Table 6. Every result is a mean over five seeds, with the complete per-seed tables in Appendix B. Code and data are released at `https://anonymous.4open.science/r/AUP`.

REFERENCES

Manaar Alam, Hithem Lamri, and Michail Maniatakos. Get rid of your trail: Remotely erasing backdoors in federated learning. *IEEE Transactions on Artificial Intelligence*, 5:6683–6698, 2024.

Andreas Athanasiou, Kangsoo Jung, and Catuscia Palamidessi. Protection against source inference attacks in federated learning. *arXiv*, 2026.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *NeurIPS*, volume 30, 2017.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *ACM SIGSAC CCS*, pp. 1175–1191, 2017.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. In *MLSys*, pp. 374–388, 2019.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *SP*, pp. 141–159, 2021.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *SP*, pp. 463–480, 2015.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *SP*, pp. 1897–1914, 2022.

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 *ICML*, pp. 4241–4268, 2023.

Jian Chen, Zehui Lin, Wanyu Lin, Wenlong Shi, Xiaoyan Yin, and Di Wang. Fedmua: Exploring the vulnerabilities of federated learning to malicious unlearning attacks. *IEEE Transactions on Information Forensics and Security*, 20:1665–1678, 2025.

Yiwei Chen, Soumyadeep Pal, Yimeng Zhang, Qing Qu, and Sijia Liu. Unlearning isn't invisible: Detecting unlearning traces in llms from model outputs. In *ICLR*, volume 2026, 2026.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009.

Jimmy Z Di, Jack Douglas, Jayadev Acharya, Gautam Kamath, and Ayush Sekhari. Hidden poison: Machine unlearning enables camouflaged poisoning attacks. In *NeurIPS*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*, 2020.

Ahmed Roushdy Elkordy, Jiang Zhang, Yahya H Ezzeldin, Konstantinos Psounis, and Salman Avestimehr. How much privacy does federated learning with secure aggregation guarantee? *arXiv*, 2022.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *ACM SIGSAC CCS*, pp. 1322–1333, 2015.

Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, Alex X Liu, and Ting Wang. Label inference attacks against vertical federated learning. In *USENIX security*, pp. 1397–1414, 2022.

Xiangshan Gao, Xingjun Ma, Jingyi Wang, Youcheng Sun, Bo Li, Shouling Ji, Peng Cheng, and Jiming Chen. Verifi: Towards verifiable federated unlearning. *IEEE Transactions on Dependable and Secure Computing*, 21:5720–5736, 2024.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? In *NeurIPS*, volume 33, 2020.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. In *NeurIPS*, volume 32, 2019.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *CVPR*, 2020.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *AAAI*, volume 35, 2021.

Hanlin Gu, Hong Xi Tae, Lixin Fan, and Chee Seng Chan. Towards privacy-guaranteed label unlearning in vertical federated learning: Few-shot forgetting without disclosure. In *ICLR*, volume 2026, 2026.

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

Elizabeth Liz Harding, Jarno J Vanto, Reece Clark, L Hannah Ji, and Sara C Ainsworth. Understanding the scope and impact of the california consumer privacy act of 2018. *Journal of Data Protection & Privacy*, 2:234–253, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv*, 2019.

Hyejun Jeong, Shiqing Ma, and Amir Houmansadr. A survey on federated unlearning: Challenges and opportunities. *IEEE Transactions on Big Data*, 2026.

Peter Kairouz and H Brendan McMahan. Advances and open problems in federated learning. *Foundations and trends in machine learning*, 14:1–210, 2021.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114: 3521–3526, 2017.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Weimin Lai, Qiao Yan, Sizhe Liang, and Kai Zhong. Federated camouflaged poisoning attack in federated unlearning. In *ICASSP*, pp. 5561–5565, 2026.

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

Oscar Li, Jiankai Sun, Xin Yang, Weihao Gao, Hongyi Zhang, Junyuan Xie, Virginia Smith, and Chong Wang. Label leakage and protection in two-party split learning. *arXiv*, 2021.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37:50–60, 2020.

Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. Federaser: Enabling efficient client-level data removal from federated learning models. In *IWQOS*, pp. 1–10, 2021.

Renyang Liu, Kangjie Chen, Han Qiu, Jie Zhang, Kwok-Yan Lam, Tianwei Zhang, and See-Kiong Ng. Saferedir: Prompt embedding redirection for robust unlearning in image generation models. *arXiv*, 2026a.

Renyang Liu, Guanlin Li, Tianwei Zhang, and See-Kiong Ng. Image can bring your memory back: A novel multi-modal guided attack against image generation model unlearning. In *International Conference on Learning Representations*, volume 2026, pp. 143251–143275, 2026b.

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 *INFOCOM*, pp. 1749–1758, 2022.

Bingguang Lu, Hongsheng Hu, Yuantian Miao, Shaleeza Sohail, Chaoxiang He, Shuo Wang, and Xiao Chen. Badfu: Backdoor federated learning through adversarial machine unlearning. In *RAID*, pp. 773–788, 2025.

Mengyao Ma, Shuofeng Liu, Minhui Xue, Surya Nepal, and Guangdong Bai. Retrace: Reinforcement learning-guided reconstruction attacks on machine unlearning. In *International Conference on Learning Representations*, volume 2026, 2026.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pp. 1273–1282, 2017.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *SP*, pp. 691–706, 2019.

Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *SP*, pp. 739–753, 2019.

Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *Transactions on Intelligent Systems and Technology*, 16:1–46, 2025.

Vardan Papyan, Xiao Y Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, volume 32, 2019.

Data Protection. General data protection regulation. *Intersoft Consulting, Accessed in October*, 24, 2018.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv*, 2020.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, pp. 3982–3992, 2019.

Thanveer Shaik, Xiaohui Tao, Haoran Xie, Lin Li, Xiaofeng Zhu, and Qing Li. Exploring the landscape of machine unlearning: A comprehensive survey and taxonomy. *IEEE Transactions on Neural Networks and Learning Systems*, 36:11676–11696, 2024.

Xinyi Sheng, Wei Bao, Hequn Wang, Yuqin Liu, and Sen Fu. Retaliatory attacks against federated unlearning via data leakage. In *AAAI*, volume 40, pp. 25321–25329, 2026.

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

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, volume 30, 2017.

Fei Wang, Baochun Li, and Bo Li. Federated unlearning and its privacy threats. *IEEE Network*, 38: 294–300, 2023.

Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Federated unlearning via class-discriminative pruning. In *WWW*, pp. 622–632, 2022.

Wei Wang, Xiangyun Tang, Yajie Wang, Yijing Lin, Tao Zhang, Meng Shen, Dusit Niyato, and Liehuang Zhu. Label inference attacks against federated unlearning. In *KSEM*, pp. 1–16, 2025a.

Wenbin Wang, Qiwen Ma, Zifan Zhang, Yuchen Liu, Zhuqing Liu, and Minghong Fang. Poisoning attacks and defenses to federated unlearning. In *WWW*, pp. 1365–1369, 2025b.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *NeurIPS*, volume 33, pp. 5776–5788, 2020.

Leijie Wu, Song Guo, Junxiao Wang, Zicong Hong, Jie Zhang, and Yaohong Ding. Federated unlearning: Guarantee the right of clients to forget. *IEEE Network*, 36:129–135, 2022.

Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, pp. 3–19, 2018.

Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of healthcare informatics research*, 5:1–19, 2021.

Fuyao Zhang, Wenjie Li, Yurong Hao, Xinyu Yan, Yang Cao, and Wei Yang Bryan Lim. Verifiably forgotten? gradient differences still enable data reconstruction in federated unlearning. *arXiv*, 2025.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Joshua C Zhao, Atul Sharma, Ahmed Roushdy Elkordy, Yahya H Ezzeldin, Salman Avestimehr, and Saurabh Bagchi. Loki: Large-scale data reconstruction attack against federated learning through model manipulation. In *SP*, pp. 1287–1305, 2024.

Lei Zhou, Youwen Zhu, and Rongke Liu. Model inversion attack against federated unlearning. *IEEE Transactions on Information Forensics and Security*, 2026.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *NeurIPS*, volume 32, 2019.

A ALGORITHMS

The compact form of AUP is Algorithm 1 in Section 4. This appendix gives the detailed per-phase pseudocode used in our implementation (Algorithms 2 and 3), followed by the two variants we evaluate (Algorithm 4). Every operation acts on the classifier-layer block delta through two primitives, RESTRICTCLF(δ), which returns only the `fc.weight` and `fc.bias` parts of a model delta, and RUNCAMPAIGN(w, a, c, s), which runs a K-round forget campaign for class c by client a from model w under seed s and returns the aggregate block delta, plus the caller's own update Δw_{local} when the server invokes it.

Algorithm 2 AUP Phase 1: calibration and per-class template construction

Require: pretrained w_0; classes $\mathcal{Y}$; probes per class N; baseline blocks B_0; detection constant k; attacker id a

record B_0 event-free blocks $\{\delta^b\}$; $\Delta_{\text{baseline}} \leftarrow \frac{1}{B_0}\sum_b \text{RESTRICTCLF}(\delta^b)$
$\sigma_{\text{base}} \leftarrow \text{std}\big(\{\,\|\text{RESTRICTCLF}(\delta^b)\|\,\}_b\big)$; $\tau \leftarrow \text{mean}\big(\{\,\|\text{RESTRICTCLF}(\delta^b)\|\,\}_b\big) + k\,\sigma_{\text{base}}$ ▷ event threshold
for each class $c \in \mathcal{Y}$ **do**
  $\mathcal{P} \leftarrow \emptyset$
  for $i = 1, \ldots, N$ **do**
    $(\Delta w, \cdot) \leftarrow \text{RUNCAMPAIGN}(w_0,\ a,\ c,\ \text{seed}(c, i))$
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{\,\text{RESTRICTCLF}(\Delta w) - \Delta_{\text{baseline}}\,\}$ ▷ baseline-centered probe
  end for
  $T_c \leftarrow \text{mean}(\mathcal{P})$ ▷ per-class template, Eq. (7)
end for
return $\Delta_{\text{baseline}},\ \tau,\ \{T_c\}_{c\in\mathcal{Y}}$

Algorithm 3 AUP Phase 2: detect-and-identify monitoring

Require: monitored block stream; $\Delta_{\text{baseline}},\ \tau,\ \{T_c\}$; forgotten count L

for each monitored block with delta δ_{obs} **do**
  if $\|\text{RESTRICTCLF}(\delta_{\text{obs}})\| > \tau$ **then** ▷ event detected
    $o \leftarrow \text{RESTRICTCLF}(\delta_{\text{obs}}) - \Delta_{\text{baseline}}$
    $\hat{Y} \leftarrow \arg\text{top-}L_{c\in\mathcal{Y}}\ \cos\big(o,\ T_c\big)$ ▷ identify, Eq. (8)
    report $\hat{Y}$
  end if
end for

Algorithm 4 Oracle and local variants: edits to the probe loop of Algorithm 2

Oracle (server-side; upper bound)

$(\Delta w,\ \Delta w_{\text{local}}) \leftarrow \text{RUNCAMPAIGN}(w_0,\ a,\ c,\ \text{seed}(c, i))$ ▷ server also receives Δw_{local}
$\mathcal{P}_{\text{loc}} \leftarrow \mathcal{P}_{\text{loc}} \cup \{\,\text{RESTRICTCLF}(\Delta w_{\text{local}})\,\}$
$T_{\text{loc},c} \leftarrow \text{mean}(\mathcal{P}_{\text{loc}})$ ▷ oracle template bank
in Phase 2, score $\cos(\text{RESTRICTCLF}(\Delta_{\text{local}}),\ T_{\text{loc},c})$ in place of $\cos(o,\ T_c)$ ▷ read the victim's own update, not the aggregate

Local (deployment; no replay from w_0)

$\Delta w \leftarrow \text{LOCALPROBE}(w_{\text{cur}},\ c)$ ▷ simulate the class-c forget campaign with the same ascent config on the attacker's own copy
average over probes as in Algorithm 2, without Δ_{baseline} subtraction ▷ a local probe carries no ambient drift to remove

The client-vantage attack never populates $\mathcal{P}_{\text{loc}}$; the oracle branch of Algorithm 4 is evaluated only to report the upper bound in Figure 2 and Appendix B. Algorithms 2 and 3 use the federation-replay

form, which restarts each probe from w_0 only for clean, reproducible per-class deltas. We additionally evaluate the local deployment variant of Algorithm 4, which builds every template on the attacker's own saved copy of the broadcast model with no federation restart and no privileged access; it decodes with the same Δ_{baseline}-centered observation $o = \Pi(\delta_{\text{obs}}) - \Delta_{\text{baseline}}$ of (8), only the template dropping Δ_{baseline} as a local probe carries no ambient drift, and it is comparable to or better than the replay form across all datasets (Appendix B).

B Full Experimental Tables

All numbers are computed over five seeds $\{1, 2, 3, 4, 5\}$ and verified against local results. The table below lists, for every single-class configuration, the full per-seed forgotten-label accuracy (exact match) with the mean $\pm$ population standard deviation, together with the Δ_{local}-oracle ceiling, the Trivial-N accuracy-drop floor, and `align.obs`, the mean cosine of the baseline-centered aggregate observation with the true-class template. `align.obs` is the realized signal-to-noise proxy of the federation-size mechanism. It broadly rises with C in every setting and its knee coincides with the accuracy knee, so the phase transition is visible directly in the observation geometry. Being a bounded cosine, `align.obs` is a monotone but uncalibrated proxy for the unbounded ratio SNR. Its transition on the axis therefore reflects the order-one SNR threshold of Proposition 2 rather than a value read off SNR itself. Because `align.obs` is the cosine of the observation with the true-class template while the decoder itself takes the largest such cosine, it is a monotone transform of the decision statistic, so we read it as a locator of the threshold rather than as an independent predictor of accuracy.

The main-text baseline table (Table 2) draws its single-class numbers from the same runs tabulated above at C=50. Trivial-N rises with C too (MNIST $0.00 \to 0.84$), so the effect is channel-level; AUP attains it without Trivial-N's labeled test set, which is why at MNIST C=200 its 0.84 tops AUP's 0.64. For the multi-class (L=3) forgetting task on CIFAR-10, AUP (replay) reaches IoU $0.30 \pm .08$, $0.53 \pm .05$, and $0.67 \pm .08$ at C=10, 50, and 100 against the random 3-class floor 0.20, tracking the single-class transition.

Local deployment variant. The tables above use the federation-replay form, which restarts each probe from w_0 for clean, reproducible per-class deltas. We also evaluate the local deployment variant (Algorithm 4), which builds every template on the attacker's own saved copy of the broadcast with no federation restart and no privileged access. At C=50 its exact match is $0.96 \pm .08$ on CIFAR-10, $0.96 \pm .08$ on CIFAR-100, $1.00 \pm .00$ on Yahoo Answers, and $0.80 \pm .18$ on MNIST, and its multi-class (L=3) IoU is $0.58 \pm .05$, above the replay 0.53. These are comparable to the replay form on CIFAR-10, CIFAR-100, and Yahoo Answers, and markedly higher on MNIST than the replay 0.56, since a local probe yields a cleaner per-class template that is not diluted by aggregation noise. The deployment variant therefore retains the attack's efficacy without any privileged rollback.

Weight coherence (AUP-free check). Mean off-diagonal cosine of `fc.weight` rows (simplex-ETF lower bound $-1/9 = -0.111$) ranges -0.109 to -0.104 on CIFAR-10 across $C \in \{10, 30, 50, 70, 100\}$ and -0.101 to -0.090 on MNIST across $C \in \{10, 30, 50, 100, 200\}$. The classifier-layer per-class direction structure AUP relies on is near-optimal and approximately constant across C, independent of absolute accuracy, a non-circular check on assumption (i) of the SNR mechanism, established without running the attack.

C Proofs

Section 5.4 summarizes these results; the full statements and proofs follow. Throughout, ϕ is the shared penultimate feature map of the broadcast model, W_k the classifier row for class k, and $p = \text{softmax}(W\phi(x) + b)$.

Notation and assumptions. Write the baseline-centered classifier-layer observation of a monitored block as $o = \alpha\, u_{c^*} + \varepsilon$, with u_{c^*} the victim's unit per-class unlearning direction, $\alpha = \|\Delta_{\text{local}}\|/\text{cpr}$ its coherent magnitude (premise (ii) of Section 3), where the $1/\text{cpr}$ weight is the IID equal-partition case of the data-weighted aggregation of (2), and ε a zero-mean, σ-subgaussian residual whose scale σ is

Table 5: Per-seed forgotten-label accuracy (exact match) for every single-class configuration (federation-replay measurement), with the Δ_{local}-oracle ceiling, the Trivial-N floor (accuracy-drop ranking on the full labeled test split), and `align.obs` (the realized signal proxy). All five seeds are shown; the last four columns are their five-seed summaries. Unlearning is gradient ascent unless the group is marked SGA-EWC.

C	seed 1	seed 2	seed 3	seed 4	seed 5	mean±std	oracle	Trivial-N	`align.obs`
CIFAR-10 / ResNet-18									
10	0.40	0.20	0.20	0.20	0.20	$0.24 \pm .08$	1.00	0.44	$+0.07$
30	0.20	0.40	0.60	0.80	0.20	$0.44 \pm .23$	1.00	0.52	$+0.13$
40	1.00	1.00	1.00	0.80	0.80	$0.92 \pm .10$	1.00	0.60	$+0.39$
50	1.00	1.00	1.00	1.00	1.00	$\mathbf{1.00 \pm .00}$	1.00	0.64	$+0.60$
70	1.00	1.00	1.00	1.00	1.00	$\mathbf{1.00 \pm .00}$	1.00	0.76	$+0.60$
100	1.00	0.80	1.00	1.00	1.00	$0.96 \pm .08$	1.00	0.68	$+0.46$
MNIST / CNN									
10	0.00	0.20	0.40	0.00	0.20	$0.16 \pm .15$	1.00	0.00	$+0.00$
30	0.40	0.40	0.00	0.40	0.40	$0.32 \pm .16$	1.00	0.28	$+0.12$
50	0.40	0.80	0.40	0.80	0.40	$0.56 \pm .20$	1.00	0.52	$+0.23$
100	0.80	0.00	0.20	0.20	0.20	$0.28 \pm .27$	1.00	0.48	$+0.09$
200	0.60	1.00	0.60	0.40	0.60	$0.64 \pm .20$	1.00	0.84	$+0.26$
CIFAR-100 / ViT-B/16									
10	0.00	0.00	0.00	0.20	0.00	$0.04 \pm .08$	1.00	0.00	-0.01
30	0.00	0.00	0.00	0.00	0.00	$0.00 \pm .00$	1.00	0.00	$+0.07$
40	0.40	0.20	0.20	0.20	0.20	$0.24 \pm .08$	1.00	0.12	$+0.14$
50	1.00	0.80	1.00	0.80	1.00	$0.92 \pm .10$	1.00	0.20	$+0.30$
70	1.00	1.00	1.00	1.00	1.00	$\mathbf{1.00 \pm .00}$	1.00	0.32	$+0.41$
100	1.00	1.00	1.00	1.00	1.00	$\mathbf{1.00 \pm .00}$	1.00	0.32	$+0.44$
Yahoo Answers / MiniLM									
10	0.60	0.80	0.80	0.80	0.80	$0.76 \pm .08$	1.00	0.20	$+0.24$
30	1.00	1.00	1.00	1.00	1.00	$\mathbf{1.00 \pm .00}$	1.00	0.32	$+0.81$
40	1.00	1.00	1.00	1.00	1.00	$\mathbf{1.00 \pm .00}$	1.00	0.32	$+0.91$
50	1.00	1.00	1.00	1.00	1.00	$\mathbf{1.00 \pm .00}$	1.00	0.36	$+0.96$
70	1.00	1.00	1.00	1.00	1.00	$\mathbf{1.00 \pm .00}$	1.00	0.44	$+0.97$
100	1.00	1.00	1.00	1.00	1.00	$\mathbf{1.00 \pm .00}$	1.00	0.44	$+0.98$
CIFAR-10 / ResNet-18, SGA-EWC unlearning									
10	0.40	0.20	0.20	0.20	0.20	$0.24 \pm .08$	1.00	0.36	$+0.07$
50	1.00	1.00	1.00	1.00	1.00	$\mathbf{1.00 \pm .00}$	1.00	0.64	$+0.55$
100	1.00	0.80	0.80	1.00	0.80	$0.88 \pm .10$	1.00	0.60	$+0.48$

Table 6: Hyperparameters. Slashed values are CIFAR-10 / MNIST / CIFAR-100 / Yahoo; all others are shared. The robustness study (Table 4) additionally varies clients per round to $\{10, 25, 40\}$ and pretraining to 250 rounds.

Setting	Value	Setting	Value
Backbone	ResNet-18 / CNN / ViT-B/16 / MiniLM		
Pretraining LR	0.05, decayed $\times 0.99$ per round, momentum 0.9		
Data partition	IID	Gradient clip (ℓ_2)	1.0
Pretraining rounds	150 / 70 / 150 / 120	SGA-EWC λ	50
Local epochs	2 / 2 / 1 / 1	Probes per class N	3 / 5 / 3 / 3
Batch size	64	Block length K	5
Clients per round	5	Baseline blocks B_0	3
Forget ascent LR	0.05	Detection constant k	3
Forget ascent epochs	5	Forget events per seed	5
Forget samples	100	Monitoring blocks	10

the dispersion of the baseline-centered event-free observation (with the scalar σ_{base} of Algorithm 2 as

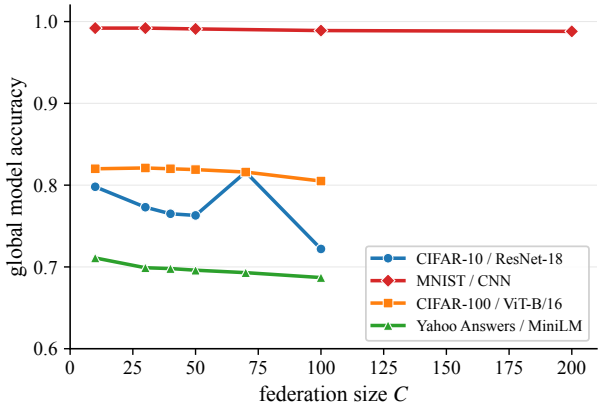

Figure 5: Global model accuracy before the forget event versus federation size C (pre-unlearning test accuracy, five-seed means). On every dataset it is flat or decreasing in C, opposite to the attack's rising trend, so the federation-size effect is not a model-quality artifact.

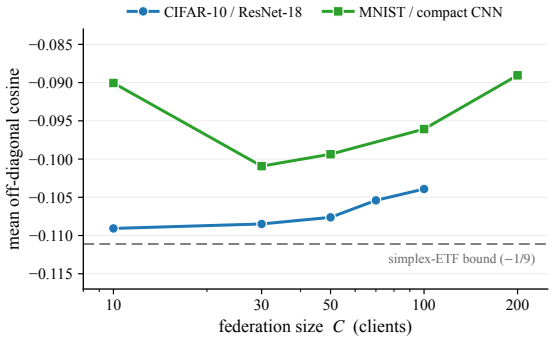

Figure 6: Classifier-row weight coherence versus federation size C (mean off-diagonal cosine of `fc.weight` rows, 5-seed means).

a proxy), so

$$\mathrm{SNR} = \frac{\|\Delta_{\mathrm{local}}\|/\mathrm{cpr}}{\sigma} = \frac{\alpha}{\sigma}. \tag{9}$$

The proofs use (i) template separation $\langle u_c, u_{c'} \rangle \le \rho < 1$ for $c \neq c'$, where ρ is the worst-case off-diagonal cosine used in the bounds, with typical values near the simplex-ETF mean $-1/(|\mathcal{Y}| - 1)$ ($\bar{\rho} \approx -0.1$ at ten classes); Figure 6 reports the mean off-diagonal cosine of the classifier rows W_c as an attack-free proxy for this template geometry; (ii) ε zero-mean, σ-subgaussian, label-independent; and (iii) attacker-victim alignment $\langle \hat{u}_c, u_c \rangle \ge 1 - \delta$, which Proposition 1 derives.

Proposition 1 (template alignment under IID, leading order)**.** *One gradient-ascent step on a class-c batch yields a classifier-layer update whose every row is a scalar multiple of the per-class mean feature $\mu_c = \mathbb{E}[\phi(x) \mid y{=}c]$, so its unit direction u_c is client-invariant to leading order, giving $\langle \hat{u}_c^A, u_c^B \rangle \ge 1 - O(1/n_c)$.*

Proof. The class-c cross-entropy gradient on row W_k is

$$\frac{\partial \mathcal{L}}{\partial W_k} = \big(p_k - \mathbf{1}[k{=}c]\big)\, \phi(x)^\top.$$

Averaging over the batch, the exact mean gradient on row W_k is $(\bar{p}_k - \mathbf{1}[k{=}c])\, \hat{\mu}_c^\top$ plus the within-batch covariance between the predictions $p_k^{(i)}$ and the features $\phi(x_i)$, which is second order in their

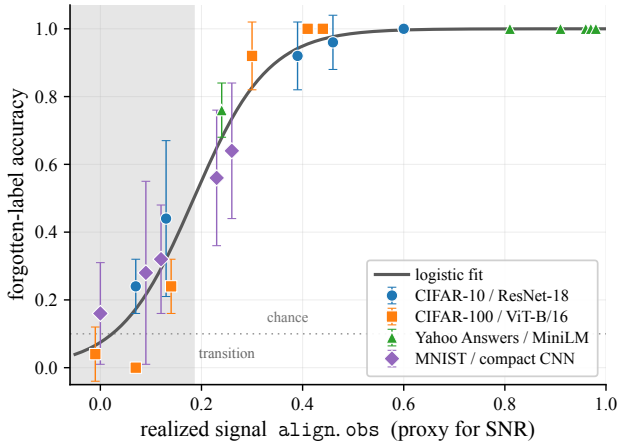

Figure 7: Identifiability transition on real data (5 seeds). Forgotten-label accuracy versus the realized signal `align.obs`, a monotone transform of the cosine decision statistic that we use to locate the transition threshold rather than as an independent predictor, with a logistic fit; error bars are one standard deviation.

fluctuations and vanishes to leading order, so

$$\Delta W_k \;\approx\; \eta\,(\bar{p}_k - \mathbf{1}[k{=}c])\,\hat{\mu}_c^{\top}, \qquad \hat{\mu}_c = \tfrac{1}{n_c}\textstyle\sum_i \phi(x_i).$$

The bias row follows the same form with the feature factor replaced by 1, $\Delta b_k \approx \eta(\bar{p}_k - \mathbf{1}[k{=}c])$, so on the classifier block $\Pi = \{\texttt{fc.weight}, \texttt{fc.bias}\}$ every row is a scalar multiple of the augmented mean $(\hat{\mu}_c, 1)$ and the unit direction u_c depends only on $(\hat{\mu}_c, \bar{p})$. Both are computed at the shared broadcast model over IID class-c data, so each client's values agree up to $O(1/\sqrt{n_c})$ sampling noise and u_c is client-invariant to leading order. An attacker A and the victim B each estimate μ_c from their own class-c data with $\|\hat{\mu}_c - \mu_c\| = O(\sigma_\phi/\sqrt{n_c})$, and since u_c is a smooth normalization of $\hat{\mu}_c$ a first-order perturbation gives

$$\langle \hat{u}_c^A,\, u_c^B \rangle \;\geq\; 1 - O(1/n_c),$$

which is assumption (iii) with $\delta = O(1/n_c)$. The deployed templates aggregate a K-round campaign, of which this single step is the leading-order term; since each step re-applies the same class-c mean-feature direction, the multi-step template stays aligned with it to the same order.

Robustness of the bound. Individual class pairs deviate from the mean, with a worst-case off-diagonal cosine around 0.2. This stays well below 1, so the margin $\alpha(1-\rho)$ remains positive and the transition of Proposition 2 holds throughout.

Proposition 2 (identifiability transition)**.** *Under Proposition 1 and (i)–(ii) with δ small, AUP returns c^* with probability at least $1 - (|\mathcal{Y}|-1)\exp(-c_0\,\alpha^2(1-\rho)^2/\sigma^2)$; identification succeeds with high probability once* $\mathrm{SNR} \gtrsim \sqrt{\log|\mathcal{Y}|}/(1-\rho)$ *and tends to* 1 *as* $\mathrm{SNR} \to \infty$*, while the accuracy falls back to the random rate* $1/|\mathcal{Y}|$ *as* $\mathrm{SNR} \to 0$.

Proof. AUP returns $\arg\max_c \langle o, \hat{u}_c \rangle$. For a competitor $c \neq c^*$ the margin splits into signal and noise,

$$M_c = \langle o,\, \hat{u}_{c^*} - \hat{u}_c \rangle \;\geq\; \underbrace{\alpha(1-\rho) - O(\sqrt{\delta})}_{\text{signal, by (i) and (iii)}} \;+\; \underbrace{\langle \varepsilon,\, \hat{u}_{c^*} - \hat{u}_c \rangle}_{\text{zero-mean, scale } \leq 2\sigma},$$

where the signal term collects the leading-order contribution and the $O(\sqrt{\delta})$ slack absorbs the attacker-victim misalignment of assumption (iii), since a unit template with $\langle \hat{u}, u \rangle \geq 1 - \delta$ deviates in norm by $O(\sqrt{\delta}) = O(1/\sqrt{n_c})$. A subgaussian tail bound followed by a union bound over the $|\mathcal{Y}| - 1$ competitors gives

$$\Pr[\hat{c} \neq c^*] \;\leq\; (|\mathcal{Y}| - 1)\,\exp\Big(-c_0\,\frac{\alpha^2(1-\rho)^2}{\sigma^2}\Big).$$

This bound is informative once $\mathrm{SNR} \gtrsim \sqrt{\log|\mathcal{Y}|}/(1-\rho)$ and vanishes as $\mathrm{SNR} \to \infty$; it is vacuous as $\mathrm{SNR} \to 0$, where the decision rule reverts to the random rate $1/|\mathcal{Y}|$ under isotropic zero-signal noise by spherical symmetry. Figure 7 confirms this on real data, where accuracy stays near chance below the threshold, saturates above it, and collapses onto a common curve across all settings up to the threshold's mild $\sqrt{\log|\mathcal{Y}|}$ growth with the class count, so the transition is a property of the signal rather than of any dataset.

Proposition 3 (the server vantage dominates the client vantage)**.** *For every decoder and every data distribution, the server-optimal identification accuracy is at least the client-optimal one,* $\mathrm{acc}^{\star}_{\textit{server}} \geq \mathrm{acc}^{\star}_{\textit{client}}$.

Proof (simulation). The broadcast is a deterministic function of the per-client updates the server stores, written here with equal weights for brevity,

$$\bar{\Delta} = \tfrac{1}{|S|}\textstyle\sum_{i\in S}\Delta_i,$$

so the server can run any client decoder g on its own reconstruction of the transcript $\{\bar{\Delta}^t\}_t$. Every client strategy is therefore available to the server,

$$\mathrm{acc}^{\star}_{\text{server}} = \max_{g'} \mathrm{acc}(g') \;\geq\; \max_{g\in\text{client}} \mathrm{acc}(g) = \mathrm{acc}^{\star}_{\text{client}},$$

with no assumption on the data distribution. The Δ_{local} oracle is one such server-side decoder, and our experiments show it attains at least AUP in every setting (Appendix B), consistent with this ceiling.

Remark. Proposition 3 ties the server-vantage attacks (ULIA, FUIA) to client-vantage AUP. Handed Δ_{local} the oracle reproduces the server bound, and without it the client vantage degrades by exactly this information gap, so aggregation alone is not a privacy floor for per-client unlearning.