MACHINE UNLEARNING UNDER AN UNCERTAIN FORGET SET VIA DISTRIBUTIONAL OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning, which aims to remove the influence of specific training samples, has received increasing attention as demands grow to remove harmful or unwanted data. Existing unlearning methods typically assume that the forget set is known exactly. However, this assumption is often impractical when detectors provide uncertain sample-wise evidence rather than a verified forget set. Under such uncertainty, prior methods reduce this evidence to retain-or-forget decisions and discard how strongly each sample should be forgotten. To address this, we propose Uncertainty-Aware Distributional Unlearning (UADU), which directly operates on continuous forgetting likelihoods. UADU (i) introduces a Likelihood-Conditioned Unlearning Objective (LCUO) that combines the training label probabilities of retain and reference-based forget targets according to forgetting likelihoods, and (ii) develops Entropy-Constrained Distributional Optimization (ECDO) to preserve the LCUO-specified training label probability and redistribute the remaining probability mass under an entropy criterion. We further provide theoretical guarantees for target approximation and local redistribution optimality. Extensive experiments on three benchmark datasets demonstrate the effectiveness and efficiency of UADU under uncertain forget sets.

1 INTRODUCTION

Modern machine learning models are often trained on large datasets that may contain harmful or incorrect samples. For example, an attacker may insert poisoned images so that an image classifier learns to associate a trigger with a target class. Once a model has learned from such samples, deleting them from the training set does not undo their effect on the model. Retraining on the remaining data can remove this effect, but often incurs substantial computational cost. Machine unlearning aims to remove the influence of selected training samples without retraining the model from scratch (Cao & Yang, 2015). The need for machine unlearning arises in diverse settings, including privacy-driven or legally mandated data deletion (Ginart et al., 2019; Chourasia & Shah, 2023), mitigation of harmful representational biases (Chen et al., 2023), and removal of the influence of mislabeled, poisoned, or backdoored training samples (Goel et al., 2024; Pawelczyk et al., 2025; Kodge et al., 2025).

Most existing unlearning methods assume a predefined forget set. They modify model parameters using gradients or weight-space information (Fan et al., 2024; Kim et al., 2025), fine-tune the model using forget–retain objectives, modified samples, or representation-level objectives (Kurmanji et al., 2023; Ebrahimpour-Boroojeny et al., 2025; Khalil et al., 2026), or provide certified removal through perturbation-based updates (Allouah et al., 2025). These methods address how to erase a known set while preserving the remaining data.

In practical settings, however, the forget set may not be known exactly and may instead need to be inferred from imperfect detection evidence. For example, detectors for mislabeled or poisoned samples provide sample-wise statistics from training dynamics or model responses (Pleiss et al., 2020; Chen et al., 2022; Pal et al., 2024). We represent the resulting forget set uncertainty using continuous forgetting likelihoods. Conventional unlearning methods that require an explicit forget set reduce these likelihoods to discrete assignments, for example through thresholding. This conversion discards continuous forgetting evidence and can assign opposite roles to samples with nearly identical likelihoods.

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.

This setting raises two coupled challenges: translating forgetting likelihoods into sample-wise unlearning behavior, and controlling the accompanying changes to the predictive distribution. We address these challenges with Uncertainty-Aware Distributional Unlearning (UADU), which comprises two components:

(i) *Likelihood-Conditioned Unlearning Objective* (LCUO) combines the training-label probabilities of the retain and reference-based forget targets according to each sample's forgetting likelihood. It constructs the forget target by weighting reference predictions with retention likelihoods and feature similarity, using orthogonal projection to exclude the training label classifier direction from the similarity computation. Thus, forgetting likelihoods condition the target prescribed by the objective rather than merely reweighting a fixed loss.

(ii) *Entropy-Constrained Distributional Optimization* (ECDO) retains the training label probability prescribed by LCUO and determines the remaining class probabilities relative to a fixed post-LCUO teacher. It preserves the teacher's entropy whenever feasible and otherwise selects the closest feasible entropy. We establish a local minimum-distortion result that justifies the redistribution used by ECDO.

Our contributions are summarized as follows:

- We formulate machine unlearning under an uncertain forget set and propose UADU to address this setting. UADU directly uses continuous forgetting likelihoods without a hard retain–forget partition.
- We develop LCUO, which defines a likelihood-conditioned unlearning objective using retention-weighted reference predictions and orthogonal feature projection, and ECDO, which determines the remaining class probabilities under training label probability and entropy constraints. In addition, we establish a target-approximation guarantee for LCUO and a local minimum-distortion result for ECDO.
- Extensive experiments across multiple datasets and unlearning settings demonstrate the effectiveness, efficiency, and robustness of UADU under uncertain forget sets.

2 Related Works

2.1 Machine Unlearning

Machine unlearning aims to make a model behave as if specified training data had never been used (Cao & Yang, 2015). Exact unlearning can incur substantial computational costs, motivating approximate methods that directly update a trained model. Representative approaches include weight scrubbing (Golatkar et al., 2020), gradient-based weight saliency (Fan et al., 2024), single-layer gradient updates (Cai et al., 2025), and adversarial-example fine-tuning (Ebrahimpour-Boroojeny et al., 2025). Complementary work establishes certified removal guarantees for convex models (Guo et al., 2020), general learning algorithms (Sekhari et al., 2021), and neural networks (Koloskova et al., 2025).

These methods generally assume a predefined forget set. Recent studies consider practical variations involving heterogeneous unlearning states or incomplete identification of forget samples. FaLW adaptively weights samples within a predefined forget set according to their unlearning states (Yu et al., 2026), whereas Cognac performs corrective graph unlearning when only part of the manipulation set is identified (Kolipaka et al., 2025). Despite these extensions, both methods still operate on discrete subsets of samples. In contrast, we study unlearning under an uncertain forget set and directly use continuous forgetting likelihoods.

2.2 Predictive Distributions for Unlearning

Predictive distributions play an important role in confidence calibration and knowledge distillation. Relevant research has examined normalized power transformations under entropy constraints (Bercher, 2012), temperature scaling for confidence calibration (Guo et al., 2017), and the separation of target-class and non-target-class terms in knowledge distillation (Zhao et al., 2022). In particular, the distillation results highlight the importance of non-target-class information for knowledge transfer.

In machine unlearning, teacher–student objectives and smoothing are used to modify predictions on forget data. SCRUB encourages agreement with the original model on retain data and disagreement on forget data (Kurmanji et al., 2023), while LoTUS smooths prediction probabilities to reduce overconfidence (Spartalis et al., 2025). Subsequent approaches include Constrained Entropic Unlearning, which flattens forget set logits subject to a retain set constraint (Entesari et al., 2026), and Attention Smoothing Unlearning, which constructs a forget teacher by increasing attention temperature and distills its predictions (Zare Zade et al., 2026). These methods define forgetting behavior for identified forget samples. When forget set membership is uncertain, however, broadly weakening a candidate sample's prediction may also suppress useful distinctions among other classes. Their formulations do not explicitly use forgetting likelihoods to control these additional changes.

3 Problem Formulation

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be a training dataset, where x_i is an input and y_i its training label. Let f_{θ_0} denote a model trained on $\mathcal{D}$. Conventional machine unlearning is given a forget set $\mathcal{D}_f \subseteq \mathcal{D}$ and seeks an updated model f_{θ_u} that removes the training influence of $\mathcal{D}_f$ while preserving predictive performance on the retain set $\mathcal{D}_r := \mathcal{D} \setminus \mathcal{D}_f$.

We study machine unlearning when the actual forget set $\mathcal{D}_f$ is not known exactly. This setting arises when detectors provide sample-wise evidence about removal candidates rather than a verified forget set. Examples include training-margin statistics for detecting mislabeled samples (Pleiss et al., 2020) and transformation sensitivity for detecting poisoned samples (Chen et al., 2022). We assume access to the original model f_{θ_0} and training dataset $\mathcal{D}$, together with a forgetting likelihood $p_i \in [0, 1]$ for each sample (x_i, y_i). Larger values indicate stronger evidence that the sample should be forgotten, and $1 - p_i$ is its retention likelihood. We treat $\{p_i\}_{i=1}^N$ as given inputs representing uncertainty in forget set membership.

This uncertainty affects both forgetting and retention. A sample with a large p_i may belong to $\mathcal{D}_r$, whereas one with a small p_i may belong to $\mathcal{D}_f$. Converting these likelihoods into a hard partition discards continuous forgetting evidence and can assign opposite unlearning roles to samples with similar likelihoods.

Although $\mathcal{D}_f$ is not known exactly, our goal is to approximate the behavior of a model trained from scratch on the actual retain set $\mathcal{D}_r$. We seek an unlearning procedure

$$\theta_u = \mathcal{U}\big(\theta_0, \mathcal{D}, \{p_i\}_{i=1}^N\big) \tag{1}$$

that operates directly on the forgetting likelihoods without retraining from scratch.

Achieving this goal raises two coupled challenges. First, the forgetting likelihood p_i indicates how strongly a sample should be forgotten, but does not specify what prediction the unlearned model should produce for that sample. Second, once this evidence is translated into a prescribed training label probability, that constraint alone does not determine how the remaining probability mass should be distributed across the other classes. We therefore need both a likelihood-conditioned specification of sample-wise unlearning behavior and a distributional mechanism that preserves this specification while controlling how the remaining probability mass is redistributed.

4 Methods

We present UADU, an uncertainty-aware unlearning framework combining LCUO and ECDO, as illustrated in Figure 1. LCUO defines a single unlearning objective conditioned on sample-wise forgetting likelihoods, without a hard retain–forget partition. After optimizing LCUO, ECDO uses the resulting model as a fixed teacher and determines the remaining class probabilities while preserving the training label probabilities prescribed by LCUO.

In Section 4.1, LCUO estimates the forget component from reference predictions weighted by retention likelihoods and feature similarity. Orthogonal projection excludes the training label classifier direction from the similarity computation. Then, we combine the retain and forget targets according to each sample's forgetting likelihood. We further analyze how closely this target approximates the expected target under the unknown retain memberships and identify the factors controlling this approximation.

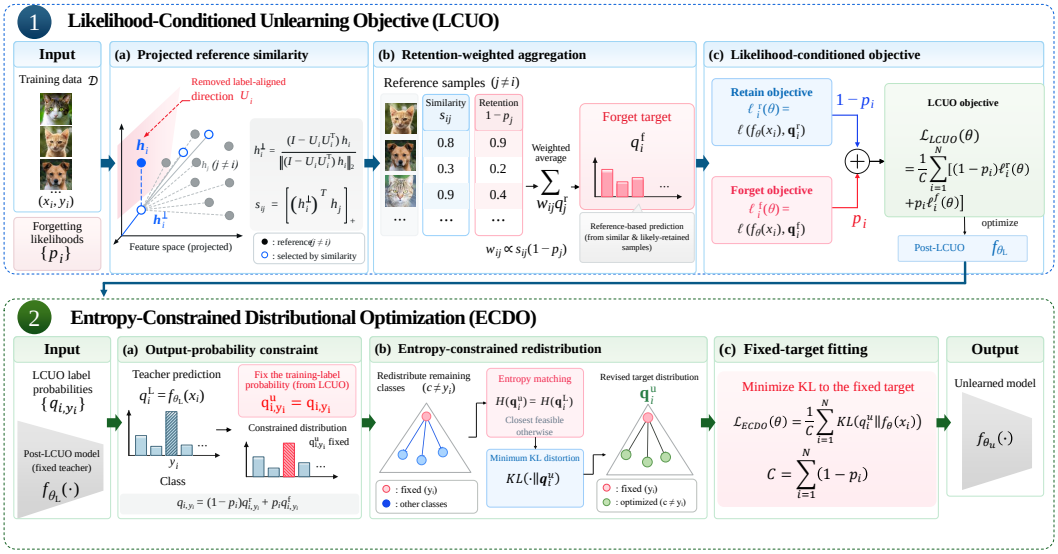


Figure 1: Overview of the UADU framework. UADU consists of two components: (1) Likelihood-Conditioned Unlearning Objective (LCUO), which constructs a reference-based forget target and combines it with the original prediction according to the forgetting likelihood. (2) Entropy-Constrained Distributional Optimization (ECDO), which preserves the LCUO-specified training label probability while redistributing the remaining probability mass under entropy and KL criteria.

In Section 4.2, ECDO determines the remaining class probabilities under the prescribed output-probability constraints. It selects the feasible entropy closest to that of a fixed post-LCUO teacher under an entropy criterion. We further show that the redistribution achieves minimum second-order KL distortion under local entropy-preserving changes. ECDO then constructs the full target distributions and fits the resulting fixed targets.

4.1 Likelihood-Conditioned Unlearning Objective

Machine unlearning aims to recover the behavior of a model trained without the forget samples, rather than merely increase their loss (Mavrothalassitis et al., 2026). We propose LCUO, which constructs a forget objective from predictions on other training samples using retention likelihoods. It combines this objective with a retain objective according to each sample's forgetting likelihood, without a hard retain–forget partition.

Let $\mathbf{q}_i^{\mathrm{r}} := f_{\theta_0}(x_i)$ denote the retain target, representing the prediction to preserve when sample i is retained. The retain objective encourages matching its training label probability q_{i,y_i}^{r}. A model retrained without the forget samples predicts those samples using information learned from the remaining data. Predictions from feature-similar samples have been shown to provide useful supervisory information (Iscen et al., 2022). In our setting, where the actual retain set is unknown, we construct the forget target $\mathbf{q}_i^{\mathrm{f}}$ from the original model's predictions on other samples $j \neq i$.

Excluding $j = i$ avoids directly reusing the sample's own prediction in its forget target. We additionally seek references that are similar to x_i beyond their shared alignment with the classifier weight for its training label y_i. We therefore exclude this direction when computing reference similarity. Let $h_i \in \mathbb{R}^d$ be the unit-normalized version of the feature vector supplied to the original model's linear classification head. Let $\mathbf{w}_{y_i} \neq 0$ denote the head's weight vector for class y_i, and set $U_i := \mathbf{w}_{y_i} / \|\mathbf{w}_{y_i}\|_2$. We compute

$$h_i^{\perp} = \frac{(I - U_i U_i^{\top}) h_i}{\|(I - U_i U_i^{\top}) h_i\|_2}, \qquad s_{ij} = [(h_i^{\perp})^{\top} h_j]_+, \tag{2}$$

where $[a]_+ = \max(a, 0)$ and the projected vector is assumed nonzero. Since $U_i^{\top} h_i^{\perp} = 0$, reference components along $\mathbf{w}_{y_i}$ do not contribute to the inner product. This projection changes the reference weights, not the predictions being averaged.

Similarity does not identify which reference samples should be retained. We therefore combine it with retention likelihood:

$$w_{ij} = \frac{s_{ij}(1-p_j)}{\sum_{k\neq i} s_{ik}(1-p_k)}, \qquad \mathbf{q}_i^{\mathrm{f}} = \sum_{j\neq i} w_{ij}\mathbf{q}_j^{\mathrm{r}}, \tag{3}$$

for $j \neq i$, assuming a positive denominator. The weights are nonnegative and sum to one. The factor $1-p_j$ reduces the direct contribution of references likely to require forgetting, without thresholding their forgetting likelihoods.

Since the retain set is not known exactly, we analyze how closely the retention-weighted aggregation in Equation (3) approximates the expected aggregation over retained references. We formalize mild assumptions. The first requires positive expected retained-reference weight, while the second controls aggregation variance.

Assumption 4.1 (Positive reference weight)**.** The total similarity weight of retained references has positive expectation.

Assumption 4.2 (Bounded aggregation variance)**.** The variance of weighted aggregation over retained references is bounded by a finite multiple of that under independent retention with the same retention probabilities.

We now compare the likelihood-weighted forget target $\mathbf{q}_i^{\mathrm{f}}$ with $\bar{\mathbf{q}}_i^{\mathrm{f}}$, the expected target obtained by applying the same averaging rule to retained references. Let ϵ_i measure retention-likelihood mismatch and β_i measure uncertainty in the total retained reference weight.

Theorem 4.3 (Retention-weighted target approximation)**.** *For any sample i satisfying Assumptions 4.1 and 4.2,*

$$\left\|\mathbf{q}_i^{\mathrm{f}} - \bar{\mathbf{q}}_i^{\mathrm{f}}\right\|_2 \leq \sqrt{2}\,\epsilon_i + 2\sqrt{2}\,\beta_i. \tag{4}$$

If the retention likelihoods are exact, the $O(\beta_i)$ rate is worst-case optimal.

Theorem 4.3 provides a theoretical justification for the reference-based forget target in Equation (3). The approximation error reflects retention-likelihood mismatch and retained-reference uncertainty, and achieves the worst-case optimal $O(\beta_i)$ rate under exact retention likelihoods. Formal definitions and the proof are provided in Appendix A.1.

We use a cross-entropy loss on the training label probability for both the retain and forget targets. Let $\ell_y(\mathbf{a}, \mathbf{b})$ denote this loss between the y-th probabilities of distributions $\mathbf{a}$ and $\mathbf{b}$. We define

$$\ell_i^{\mathrm{r}}(\theta) = \ell_{y_i}(f_\theta(x_i), \mathbf{q}_i^{\mathrm{r}}), \qquad \ell_i^{\mathrm{f}}(\theta) = \ell_{y_i}\left(f_\theta(x_i), \mathbf{q}_i^{\mathrm{f}}\right). \tag{5}$$

LCUO combines these losses according to each forgetting likelihood, with $C = \sum_{i=1}^{N}(1-p_i)$:

$$\mathcal{L}_{\mathrm{LCUO}}(\theta) = \frac{1}{C}\sum_{i=1}^{N}\left[(1-p_i)\ell_i^{\mathrm{r}}(\theta) + p_i\ell_i^{\mathrm{f}}(\theta)\right]. \tag{6}$$

Since the cross-entropy loss is affine in its target probability, Equation (6) is equivalent to matching the training label probability to

$$\widetilde{q}_{i,y_i} = (1-p_i)q_{i,y_i}^{\mathrm{r}} + p_i q_{i,y_i}^{\mathrm{f}}. \tag{7}$$

Thus, as p_i increases, the training label target moves continuously from the original prediction toward the reference-based forget target, without a hard retain–forget partition. We optimize Equation (6) from θ_0 to obtain $f_{\theta_{\mathrm{L}}}$. ECDO then determines the remaining class probabilities.

4.2 Entropy-Constrained Distributional Optimization

After LCUO, we obtain the target training label probability $\widetilde{q}_{i,y_i}$ and the post-LCUO prediction $\mathbf{q}_i^{\mathrm{L}} := f_{\theta_{\mathrm{L}}}(x_i)$. ECDO preserves $\widetilde{q}_{i,y_i}$ while determining the remaining class probabilities relative to $\mathbf{q}_i^{\mathrm{L}}$. We explain how entropy preservation controls this redistribution and characterize its local KL distortion.

Let $\mathbf{q}_i^{\mathrm{u}}$ denote the full target distribution constructed by ECDO. Its training label probability is constrained by

$$q_{i,y_i}^{\mathrm{u}} = \widetilde{q}_{i,y_i} = (1-p_i)q_{i,y_i}^{\mathrm{r}} + p_i q_{i,y_i}^{\mathrm{f}}. \tag{8}$$

The training label identifies the class that the model is trained to predict, while the other coordinates determine how the remaining probability mass is distributed across classes. Here, $\widetilde{q}_{i,y_i}$ is the likelihood-conditioned training label target obtained by combining the retain and forget targets according to p_i.

Optimizing the LCUO objective through shared model parameters yields the prediction $\mathbf{q}_i^{\mathrm{L}} := f_{\theta_{\mathrm{L}}}(x_i)$, whose training label probability need not equal $\widetilde{q}_{i,y_i}$. We retain $\widetilde{q}_{i,y_i}$ as the explicit constraint inherited from LCUO and use $\mathbf{q}_i^{\mathrm{L}}$ as a reference for the remaining probabilities. If $q_{i,y_i}^{\mathrm{L}} = \widetilde{q}_{i,y_i}$, the reference already satisfies this constraint and no redistribution is required.

Otherwise, fixing $\widetilde{q}_{i,y_i}$ leaves probability mass $1-\widetilde{q}_{i,y_i}$ for the other classes. Their relative probabilities must then be determined. Keeping the relative probabilities of $\mathbf{q}_i^{\mathrm{L}}$ unchanged is the simplest choice, but it can change the entropy of the full prediction. Prior work has shown that predictive distributions can retain informative structure beyond the most likely class (Müller et al., 2019; Zhao et al., 2022). Let $H(\mathbf{q}) := -\sum_c q_c \log q_c$ denote the Shannon entropy of a predictive distribution. We use $H(\mathbf{q}_i^{\mathrm{L}})$ as a reference for predictive uncertainty, preserving it whenever feasible and otherwise selecting the closest feasible entropy. Since entropy alone does not uniquely determine the remaining probabilities, we also consider how the target responds to changes in the training label probability. With the reference fixed, we seek entropy-preserving adjustments with minimum local KL distortion.

To characterize this redistribution, define the conditional distribution of $\mathbf{q}_i^{\mathrm{L}}$ over classes $c \neq y_i$:

$$u_{i,c} = \frac{q_{i,c}^{\mathrm{L}}}{1-q_{i,y_i}^{\mathrm{L}}}, \qquad c \neq y_i. \tag{9}$$

Keeping the relative non-label probabilities unchanged while rescaling them to $1-\widetilde{q}_{i,y_i}$ generally changes the full entropy. ECDO therefore adjusts their relative proportions using the following normalized power transformation:

$$q_{i,y_i}^{\mathrm{u}} = \widetilde{q}_{i,y_i}, \qquad q_{i,c}^{\mathrm{u}} = (1-\widetilde{q}_{i,y_i})\frac{u_{i,c}^{\alpha_i}}{\sum_{k\neq y_i} u_{i,k}^{\alpha_i}}, \qquad c \neq y_i. \tag{10}$$

The transformation preserves the ordering and ties among the non-label classes while controlling their concentration. We choose α_i to match $H(\mathbf{q}_i^{\mathrm{L}})$ whenever feasible, and otherwise to attain the closest feasible entropy.

To justify the redistribution in Equation (10), we study how the remaining class probabilities should change when the training label probability varies while the reference entropy is preserved. We consider the standard softmax setting with nonuniform probabilities across the remaining classes.

Theorem 4.4 (Minimum entropy-preserving distortion)**.** *Given a local change in the training label probability, Equation (10) induces the unique entropy-preserving adjustment minimizing the second-order term of* $\mathrm{KL}(\cdot\|\mathbf{q}_i^{\mathrm{u}})$.

Thus, once LCUO specifies the training label probability, ECDO provides a locally least-distorting redistribution of the remaining probability mass under entropy preservation. Formal details and the proof are provided in Appendix A.2.

Finally, starting from θ_{L}, we fit the fixed targets using

$$\mathcal{L}_{\mathrm{ECDO}}(\theta) = \frac{1}{C}\sum_{i=1}^{N} \mathrm{KL}(\mathbf{q}_i^{\mathrm{u}}\|f_\theta(x_i)), \qquad C = \sum_{i=1}^{N}(1-p_i) > 0. \tag{11}$$

This KL objective decomposes into a training-label probability term and a conditional KL term over the remaining classes. Let θ_u denote the resulting parameters and f_{θ_u} the final unlearned model.

As a result, UADU operates directly on continuous forgetting likelihoods without requiring a hard retain–forget partition. LCUO determines the likelihood-conditioned training label targets from retention-weighted references, while ECDO preserves these probabilities and determines the remaining target probabilities under entropy and KL criteria.

5 EXPERIMENTS

In this section, we evaluate UADU under uncertain forget sets across multiple benchmark datasets. We further conduct ablation studies to examine the contribution of each component. Additionally, we evaluate UADU with different forgetting likelihood detectors and under an alternative unlearning setting to further assess the robustness and generality of our framework.

5.1 EXPERIMENTAL SETUPS

Datasets and Tasks. We conduct image classification experiments on CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), and TinyImageNet (Le & Yang, 2015). Following prior machine unlearning studies (Fan et al., 2024), we adopt ResNet-18 (He et al., 2016) as the backbone for all three datasets. We consider a mislabeled-data unlearning setting, where the model is trained on data containing incorrect labels and the true forget set is not known exactly. We inject 10% symmetric label noise and use the corrupted samples as the ground-truth forget set for evaluation. We obtain forgetting likelihoods using Area Under the Margin (AUM), proposed for mislabel detection (Pleiss et al., 2020). UADU directly uses the continuous forgetting likelihoods $\{p_i\}_{i=1}^N$, whereas baselines requiring explicit retain and forget sets construct $\widehat{\mathcal{D}}_f(\tau) = \{(x_i, y_i) \in \mathcal{D} : p_i \geq \tau\}$ and $\widehat{\mathcal{D}}_r(\tau) = \mathcal{D} \setminus \widehat{\mathcal{D}}_f(\tau)$. We use $\tau = 0.5$ in the main experiments. Further implementation details are provided in Appendix B.

Baseline methods. We compare UADU with the following unlearning methods: (1) **Retrain** refers to training a model from scratch on the ground-truth retain data only. (2) **Fine-Tuning (FT)** (Warnecke et al., 2021) fine-tunes the original model on the retain set. (3) **Gradient Ascent (GA)** (Thudi et al., 2022) maximizes the loss on the forget set. (4) **LAF+R** (Shen et al., 2024) performs representation-level forgetting and alignment, followed by supervised repairing. (5) **SalUn** (Fan et al., 2024) identifies salient model parameters using forget gradients and performs random-label unlearning over the selected parameters. (6) **MUNBa** (Wu & Harandi, 2025) formulates forgetting and preservation as a Nash bargaining problem to obtain a balanced gradient direction. (7) **AMUN** (Ebrahimpour-Boroojeny et al., 2025) fine-tunes the model using adversarial examples of the forget samples to reduce their prediction confidence. For all comparison methods, we follow the hyperparameter settings reported in the original papers and their publicly available implementations as closely as possible.

Evaluation Metrics. We evaluate all methods using four metrics commonly adopted in machine unlearning (Jia et al., 2023; Fan et al., 2024). Forget accuracy (A_f) and retain accuracy (A_r) are measured on the ground-truth forget and retain subsets of the training data, respectively, with both expected to approach Retrain. Membership inference attack (MIA) measures the residual membership information of the forget samples, following the logistic-regression-based evaluation used in prior work (Chundawat et al., 2023; Foster et al., 2024). Execution time (ET) measures the computational cost of producing the unlearned model. Our objective is to produce an unlearned model that closely matches the behavior of the retrained model at lower computational cost.

Table 1: Main unlearning results on CIFAR-10, CIFAR-100, and TinyImageNet. Results are averaged over three random seeds. For A_r, A_f, and MIA, results closer to Retrain are better, while lower ET is better. **Bold** indicates the best result and underlining indicates the second best result among unlearning methods.

Method	CIFAR-10				CIFAR-100				TinyImageNet			
	A_r (%)	A_f (%)	MIA (%)	ET (s)	A_r (%)	A_f (%)	MIA (%)	ET (s)	A_r (%)	A_f (%)	MIA (%)	ET (s)
Original	99.95	99.25	66.81	–	99.97	99.88	71.49	–	99.98	99.97	67.82	–
Retrain	$100.00_{\pm 0.00}$	$0.67_{\pm 0.06}$	$75.75_{\pm 0.48}$	344.07	$99.98_{\pm 0.00}$	$0.23_{\pm 0.01}$	$44.07_{\pm 0.28}$	345.98	$99.98_{\pm 0.00}$	$0.18_{\pm 0.02}$	$33.59_{\pm 0.76}$	2923.73
FT	$99.82_{\pm 0.08}$	$38.24_{\pm 2.13}$	$14.07_{\pm 1.99}$	37.29	**99.97**$_{\pm 0.00}$	$95.53_{\pm 0.50}$	$14.90_{\pm 1.56}$	27.32	$\underline{99.03}_{\pm 0.26}$	$54.94_{\pm 2.42}$	$13.06_{\pm 1.11}$	263.45
GA	**99.89**$_{\pm 0.09}$	$43.25_{\pm 5.19}$	$\underline{43.18}_{\pm 5.77}$	**4.26**	**99.97**$_{\pm 0.00}$	$99.83_{\pm 0.01}$	$76.08_{\pm 1.53}$	**9.92**	**99.98**$_{\pm 0.00}$	$99.88_{\pm 0.02}$	$54.92_{\pm 3.45}$	**43.39**
MUNBa	$\underline{99.85}_{\pm 0.03}$	$31.05_{\pm 1.39}$	$14.91_{\pm 0.95}$	45.34	$\underline{96.94}_{\pm 1.36}$	$50.61_{\pm 1.92}$	$12.42_{\pm 0.94}$	59.30	$97.85_{\pm 0.85}$	$39.96_{\pm 3.29}$	$10.18_{\pm 0.96}$	429.99
LAF+R	$97.23_{\pm 0.65}$	$13.42_{\pm 0.90}$	$9.13_{\pm 0.70}$	419.35	$80.57_{\pm 8.05}$	$\underline{20.34}_{\pm 12.68}$	$\underline{29.74}_{\pm 2.44}$	320.40	$96.58_{\pm 2.39}$	$55.54_{\pm 28.07}$	$\underline{28.89}_{\pm 11.81}$	1459.75
SalUn	$99.26_{\pm 0.09}$	$\underline{12.82}_{\pm 1.43}$	$34.87_{\pm 4.65}$	42.87	$87.18_{\pm 2.45}$	$28.15_{\pm 3.48}$	$3.20_{\pm 0.62}$	110.14	$90.91_{\pm 2.71}$	$\underline{21.20}_{\pm 4.04}$	$6.73_{\pm 0.77}$	641.81
AMUN	$98.91_{\pm 0.04}$	$55.91_{\pm 1.11}$	$14.90_{\pm 0.42}$	69.96	$95.41_{\pm 0.38}$	$72.34_{\pm 1.42}$	$28.03_{\pm 1.94}$	96.89	$95.69_{\pm 1.05}$	$45.97_{\pm 1.47}$	$22.47_{\pm 1.43}$	466.56
UADU (ours)	$98.55_{\pm 0.13}$	**5.93**$_{\pm 0.47}$	**51.01**$_{\pm 2.49}$	$\underline{12.98}$	$90.21_{\pm 1.81}$	**5.62**$_{\pm 0.76}$	**33.58**$_{\pm 1.20}$	$\underline{13.07}$	$93.28_{\pm 1.07}$	**15.04**$_{\pm 2.72}$	**30.66**$_{\pm 2.42}$	$\underline{172.07}$

5.2 Comparison with Baselines

Table 1 presents the comparison with baseline methods across the three datasets. (1) UADU achieves the A_f closest to Retrain across all three datasets. Several baselines attain A_r closer to Retrain, but their substantially higher A_f indicates that this retention is accompanied by limited forgetting. UADU instead brings the forget set behavior substantially closer to Retrain while maintaining high retain set accuracy. (2) UADU also yields the MIA results closest to Retrain across all three datasets. This alignment shows that UADU moves the membership behavior closer to that of Retrain, providing complementary evidence of effective unlearning beyond predictive accuracy.

UADU achieves the second-lowest ET on every dataset and is substantially faster than full retraining. Compared with the average ET of the baselines within each dataset, UADU reduces execution time by 81.2% on average across the three datasets. This consistent efficiency demonstrates that UADU performs unlearning under uncertain forget sets with substantially lower computational cost. Together with its close agreement with Retrain in A_f and MIA, these results demonstrate that UADU achieves effective unlearning under uncertain forget sets at substantially lower computational cost than full retraining.

5.3 Ablation Study

We conduct ablation experiments on CIFAR-10 and CIFAR-100 to examine the contributions of the key components of UADU. Specifically, *w/o LCUO optimization* represents UADU without LCUO optimization, while retaining its target construction. *w/o conditional KL* represents UADU without the conditional KL term in ECDO fitting, using only the training label probability-matching loss. *w/o entropy constraint* represents UADU without the entropy-preserving constraint. Figure 2 reports the resulting A_f and MIA gap.

The results of A_f and MIA gap on CIFAR-10 and CIFAR-100, shown in Figure 2, provide three insights. (1) Removing LCUO optimization substantially increases A_f on both datasets, showing that LCUO optimization is important for realizing the prescribed forgetting behavior. (2) Removing the conditional KL objective increases both A_f and the MIA gap, particularly on CIFAR-100, showing that controlling the non-label redistribution contributes to the final unlearning behavior. (3) Removing the entropy constraint leaves A_f nearly unchanged but consistently enlarges the MIA gap, indicating that entropy preservation mainly controls the redistribution of the remaining probability mass rather than the strength of forgetting. Together, these results distinguish the roles of the components: LCUO determines the forgetting behavior, while the conditional KL objective and entropy constraint regulate the resulting predictive distribution.

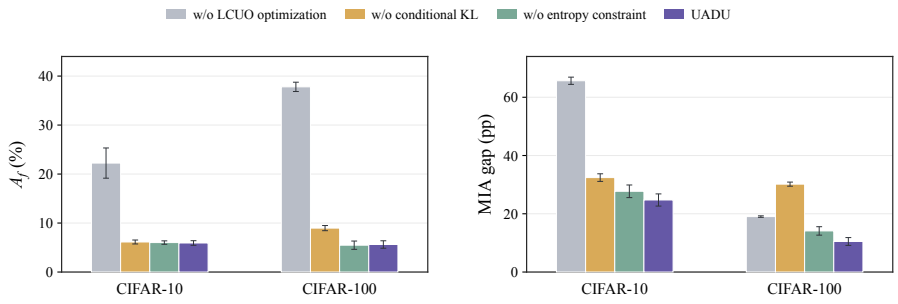


Figure 2: Ablation results on CIFAR-10 and CIFAR-100. We report forget accuracy (A_f) and the MIA gap relative to Retrain. Lower values are better for both metrics.

5.4 Robustness Across Unlearning Settings

We extend the evaluation on CIFAR-10 to test whether UADU depends on the detector or unlearning setting used in the main experiments. Under mislabeled-data unlearning, we replace AUM with EL2N (Paul et al., 2021). We further consider backdoor unlearning under a clean-label attack (Barni et al., 2019), using STRIP (Gao et al., 2019) to obtain sample-wise poisoning evidence.

Table 2: Additional results under backdoor and mislabeled-data unlearning on CIFAR-10.

	Backdoor Unlearning			Mislabeled-Data Unlearning		
Method	A_r (%)	A_f (%)	MIA (%)	A_r (%)	A_f (%)	MIA (%)
Original	100.00	100.00	99.16	99.95	99.25	66.81
Retrain	100.00	34.80	10.37	100.00	0.67	75.75
FT	**99.99**	100.00	99.68	99.74	45.59	19.60
GA	90.63	4.96	40.27	99.76	45.02	44.59
MUNBa	99.92	100.00	99.77	**99.87**	39.43	19.30
LAF+R	99.03	0.27	62.84	95.55	15.89	23.37
SalUn	99.96	100.00	98.84	99.12	17.33	38.31
AMUN	99.88	99.77	87.91	98.33	60.23	18.23
UADU (ours)	99.69	**20.49**	**0.08**	97.60	**10.90**	**50.42**

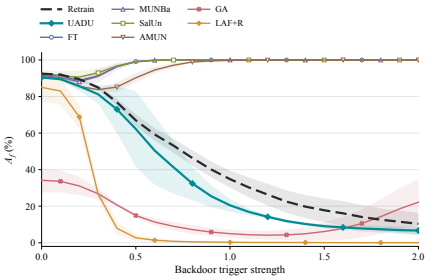


Figure 3: Forget accuracy across different backdoor strengths on CIFAR-10.

In both settings, detector outputs are converted into continuous forgetting likelihoods. Further details are provided in Appendix B.

Table 2 reports results under the two additional settings. The left block considers backdoor unlearning with STRIP-based forgetting likelihoods, while the right block keeps the mislabeled-data setting and replaces AUM with EL2N. UADU achieves the A_f and MIA closest to Retrain in both settings, while retaining A_r values of 99.69% and 97.60%, respectively. This indicates that its effectiveness is not tied to the AUM-based forgetting evidence used in the main experiments.

Figure 3 further examines the backdoor setting by varying the backdoor trigger strength and measuring the resulting forget accuracy A_f of each unlearning method on CIFAR-10. As the trigger strength changes, the A_f of Retrain also changes substantially, showing that the desired forgetting behavior is not a fixed reduction toward zero. Several baselines either remain close to the Original model or rapidly drive A_f toward zero, deviating from Retrain in opposite directions. In contrast, UADU more closely follows the changing Retrain trend across trigger strengths. This result shows that UADU maintains retraining-like behavior under uncertain forget sets, where the evidence supporting sample removal varies across conditions. This supports its effectiveness in adapting unlearning behavior to uncertain forgetting evidence.

5.5 Additional Experiments

To further assess the robustness of UADU, we provide additional experiments in Appendix C, including sensitivity to the forget set size, optimization hyperparameters, baseline threshold, and model architecture. Appendix D further provides empirical examinations of the theoretical properties established for LCUO and ECDO.

6 Conclusion

We study machine unlearning when the forget set is not known exactly and is represented by sample-wise forgetting likelihoods. We propose UADU, which uses these likelihoods directly rather than reducing them to a hard retain–forget partition. LCUO constructs likelihood-conditioned unlearning targets from retention-weighted reference predictions, while ECDO preserves the LCUO-specified training label probability and determines the remaining predictive distribution through entropy-constrained redistribution. Our analysis provides a target-approximation guarantee for LCUO and a local minimum-distortion characterization for ECDO.

Extensive experiments on multiple benchmark datasets demonstrate the effectiveness and efficiency of UADU under uncertain forget sets. Additional evaluations support its robustness, while numerical results are consistent with the theoretical properties of LCUO and ECDO. UADU currently treats forgetting likelihoods as externally supplied inputs. Future work could jointly estimate and calibrate these likelihoods within the unlearning process.

REFERENCES

Youssef Allouah, Joshua Kazdan, Rachid Guerraoui, and Sanmi Koyejo. The utility and complexity of in-and out-of-distribution machine unlearning. In *International Conference on Learning Representations*, volume 2025, pp. 2713–2738, 2025.

Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International conference on machine learning*, pp. 312–321. PMLR, 2019.

Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 101–105. IEEE, 2019.

J-F Bercher. A simple probabilistic construction yielding generalized entropies and divergences, escort distributions and q-gaussians. *Physica A: Statistical Mechanics and its Applications*, 391 (19):4460–4469, 2012.

Zikui Cai, Yaoteng Tan, and M. Salman Asif. Targeted unlearning with single layer unlearning gradient. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 6257–6290. PMLR, 13–19 Jul 2025.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pp. 463–480, 2015. doi: 10.1109/SP.2015.35.

Ruizhe Chen, Jianfei Yang, Huimin Xiong, Jianhong Bai, Tianxiang Hu, Jin Hao, Yang Feng, Joey Tianyi Zhou, Jian Wu, and Zuozhu Liu. Fast model debias with machine unlearning. *Advances in Neural Information Processing Systems*, 36:14516–14539, 2023.

Weixin Chen, Baoyuan Wu, and Haoqian Wang. Effective backdoor defense by exploiting sensitivity of poisoned samples. *Advances in Neural Information Processing Systems*, 35:9727–9737, 2022.

Rishav Chourasia and Neil Shah. Forget unlearning: Towards true data-deletion in machine learning. In *International conference on machine learning*, pp. 6028–6073. PMLR, 2023.

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7210–7217, 2023.

Ali Ebrahimpour-Boroojeny, Hari Sundaram, and Varun Chandrasekaran. Not all wrong is bad: Using adversarial examples for unlearning. In *Forty-second International Conference on Machine Learning*, 2025.

Taha Entesari, Arman Hatami, Rinat Khaziev, Anil Ramakrishna, and Mahyar Fazlyab. Constrained entropic unlearning: A primal-dual framework for large language models. *Advances in Neural Information Processing Systems*, 38:166435–166474, 2026.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *International Conference on Learning Representations*, volume 2024, pp. 53643–53673, 2024.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 12043–12051, 2024.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, pp. 113–125, 2019.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019.

Shashwat Goel, Ameya Prabhu, Philip Torr, Ponnurangam Kumaraguru, and Amartya Sanyal. Corrective machine unlearning. *Transactions on Machine Learning Research*, 2024.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9301–9309. IEEE, 2020.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3832–3842. PMLR, 13–18 Jul 2020.

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*, pp. 770–778, 2016.

Ahmet Iscen, Jack Valmadre, Anurag Arnab, and Cordelia Schmid. Learning with neighbor consistency for noisy labels. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4662–4671. IEEE, 2022.

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsity can simplify machine unlearning. *Advances in neural information processing systems*, 36:51584–51605, 2023.

Yasser Khalil, Mehdi Setayesh, and Hongliang Li. Coun: Empowering machine unlearning via contrastive learning. *Advances in Neural Information Processing Systems*, 38:131254–131283, 2026.

Hyo Seo Kim, Dongyoon Han, and Junsuk Choe. Negmerge: Sign-consensual weight merging for machine unlearning. In *Forty-second International Conference on Machine Learning*, 2025.

Sangamesh Kodge, Deepak Ravikumar, Gobinda Saha, and Kaushik Roy. Sap: corrective machine unlearning with scaled activation projection for label noise robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17930–17937, 2025.

Varshita Kolipaka, Akshit Sinha, Debangan Mishra, Sumit Kumar, Arvindh Arun, Shashwat Goel, and Ponnurangam Kumaraguru. A cognac shot to forget bad memories: Corrective unlearning for graph neural networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31223–31248. PMLR, 2025.

Anastasia Koloskova, Youssef Allouah, Animesh Jha, Rachid Guerraoui, and Sanmi Koyejo. Certified unlearning for neural networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31275–31298. PMLR, 13–19 Jul 2025.

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

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36:1957–1987, 2023.

Y. Le and X. Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Lucien LeCam. Convergence of estimates under dimensionality restrictions. *The Annals of Statistics*, pp. 38–53, 1973.

Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

Ioannis Mavrothalassitis, Pol Puigdemont, Noam Levi, and Volkan Cevher. Ascent fails to forget. *Advances in Neural Information Processing Systems*, 38:45330–45365, 2026.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

J Nocedal and S Wright. Numerical optimization, 2006.

Soumyadeep Pal, Yuguang Yao, Ren Wang, Bingquan Shen, and Sijia Liu. Backdoor secrets unveiled: Identifying backdoor data with optimized scaled prediction consistency. In *International conference on learning representations*, volume 2024, pp. 14061–14083, 2024.

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. *Advances in neural information processing systems*, 32, 2019.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in neural information processing systems*, 34:20596–20607, 2021.

Martin Pawelczyk, Jimmy Z. Di, Yiwei Lu, Gautam Kamath, Ayush Sekhari, and Seth Neel. Machine unlearning fails to remove data poisoning attacks. In *The Thirteenth International Conference on Learning Representations*, 2025.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33:17044–17056, 2020.

Xiangyu Qi, Tinghao Xie, Jiachen T Wang, Tong Wu, Saeed Mahloujifar, and Prateek Mittal. Towards a proactive {ML} approach for detecting backdoor poison samples. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 1685–1702, 2023.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021.

Shaofei Shen, Chenhao Zhang, Yawen Zhao, Alina Bialkowski, Weitong Chen, and Miao Xu. Label-agnostic forgetting: A supervision-free unlearning in deep models. In *International Conference on Learning Representations*, volume 2024, pp. 54217–54235, 2024.

Christoforos N Spartalis, Theodoros Semertzidis, Efstratios Gavves, and Petros Daras. Lotus: Large-scale machine unlearning with a taste of uncertainty. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10046–10055. IEEE, 2025.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European symposium on security and privacy (EuroS&P)*, pp. 303–319. IEEE, 2022.

Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. *arXiv preprint arXiv:2108.11577*, 2021.

Jing Wu and Mehrtash Harandi. Munba: Machine unlearning via nash bargaining. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–12. IEEE, 2025.

Liheng Yu, Zhe Zhao, Yuxuan Wang, Pengkun Wang, Xiaofeng Cao, Binwu Wang, and Yang Wang. Falw: A forgetting-aware loss reweighting for long-tailed unlearning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Saleh Zare Zade, Xiangyu Zhou, Sijia Liu, and Dongxiao Zhu. Attention smoothing is all you need for unlearning. In *International Conference on Learning Representations*, volume 2026, pp. 19583–19621, 2026.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11943–11952. IEEE, 2022.

A PROOFS

A.1 PROOF OF THEOREM 4.3

Proof. For any sample i satisfying Assumptions 4.1 and 4.2, condition on the available evidence and on forgetting i. All expectations and covariances are taken under this conditioning, all sums exclude i, and the reference quantities are fixed. Let $M_j \in \{0,1\}$ indicate whether reference j is retained and let $\pi_j = \mathbb{E}[M_j]$. Writing $\mathbf{M} = (M_j)_{j\neq i}$, $\Sigma_i = \mathrm{Cov}(\mathbf{M})$, and $D_i = \mathrm{diag}(\pi_j(1-\pi_j))$, Assumption 4.2 gives $\Sigma_i \preceq \kappa_i D_i$ for some finite $\kappa_i \geq 1$. Define

$$\epsilon_i = \frac{\sum_j s_{ij}|(1-p_j)-\pi_j|}{\sum_j s_{ij}(1-p_j)}, \qquad \beta_i = \frac{\kappa_i \sum_j s_{ij}^2 \pi_j(1-\pi_j)}{\Big(\sum_j s_{ij}\pi_j\Big)^2}.$$

Let $\mathbf{Q} = \sum_j s_{ij} M_j \mathbf{q}_j^{\mathrm{r}} / \sum_j s_{ij} M_j$ when the denominator is positive, and set $\mathbf{Q} = \mathbf{q}_i^{\mathrm{r}}$ otherwise, so that $\bar{\mathbf{q}}_i^{\mathrm{f}} = \mathbb{E}[\mathbf{Q}]$. Also define

$$\mathbf{q}_i^{\pi} = \frac{\sum_j s_{ij}\pi_j \mathbf{q}_j^{\mathrm{r}}}{\sum_j s_{ij}\pi_j}, \qquad S = \frac{\sum_j s_{ij} M_j}{\sum_j s_{ij}\pi_j}.$$

Step 1: Likelihood mismatch. Since $\sum_j s_{ij}\pi_j(\mathbf{q}_j^{\mathrm{r}} - \mathbf{q}_i^{\pi}) = 0$ and probability vectors have Euclidean diameter $\sqrt{2}$, direct subtraction gives

$$\big\|\mathbf{q}_i^{\mathrm{f}} - \mathbf{q}_i^{\pi}\big\|_2 = \left\| \frac{\sum_j s_{ij}\big((1-p_j)-\pi_j\big)(\mathbf{q}_j^{\mathrm{r}} - \mathbf{q}_i^{\pi})}{\sum_j s_{ij}(1-p_j)} \right\|_2 \leq \sqrt{2}\,\epsilon_i.$$

Step 2: Random normalization. Multiplying by S removes the random denominator and gives

$$S(\mathbf{Q} - \mathbf{q}_i^{\pi}) = \frac{\sum_j s_{ij}(M_j - \pi_j)(\mathbf{q}_j^{\mathrm{r}} - \mathbf{q}_i^{\pi})}{\sum_j s_{ij}\pi_j}.$$

Hence, $\mathbb{E}[S] = 1$ and $\mathbb{E}[S(\mathbf{Q} - \mathbf{q}_i^{\pi})] = 0$. Applying $\Sigma_i \preceq \kappa_i D_i$ to S and coordinatewise to the centered sum gives

$$\mathrm{Var}(S) \leq \beta_i,$$

$$\mathbb{E}\big[S^2 \|\mathbf{Q} - \mathbf{q}_i^{\pi}\|_2^2\big] \leq \frac{\kappa_i \sum_j s_{ij}^2 \pi_j(1-\pi_j)\|\mathbf{q}_j^{\mathrm{r}} - \mathbf{q}_i^{\pi}\|_2^2}{\Big(\sum_j s_{ij}\pi_j\Big)^2} \leq 2\beta_i.$$

Using $1 - S = (S-1)^2 - S(S-1)$ and Cauchy–Schwarz (Nocedal & Wright, 2006),

$$\begin{aligned}
\|\bar{\mathbf{q}}_i^{\mathrm{f}} - \mathbf{q}_i^{\pi}\|_2 &= \|\mathbb{E}[(1-S)(\mathbf{Q} - \mathbf{q}_i^{\pi})]\|_2 \\
&\leq \mathbb{E}[(S-1)^2 \|\mathbf{Q} - \mathbf{q}_i^{\pi}\|_2] + \mathbb{E}[|S-1|S\|\mathbf{Q} - \mathbf{q}_i^{\pi}\|_2] \\
&\leq \sqrt{2}\ \mathrm{Var}(S) + \sqrt{\mathrm{Var}(S)\mathbb{E}[S^2\|\mathbf{Q} - \mathbf{q}_i^{\pi}\|_2^2]} \\
&\leq 2\sqrt{2}\,\beta_i.
\end{aligned}$$

Combining this with Step 1 proves the upper bound.

Step 3: Worst-case rate optimality. We now show that the $O(\beta_i)$ dependence cannot generally be improved using only sample-wise retention information. Following a two-point indistinguishability argument (LeCam, 1973), we construct two retention mechanisms with identical retention marginals and the same dependence bound, but whose expected retained-reference targets differ by order β_i. Any estimator using only this information must therefore incur $\Omega(\beta_i)$ error under at least one of the two mechanisms.

For any $n \geq 1$, consider n always-retained references with prediction $\mathbf{u}$ and two uncertain references with prediction $\mathbf{v}$, all with equal positive similarities, where $\|\mathbf{u} - \mathbf{v}\|_2 = 1/\sqrt{2}$. Each uncertain reference has retention probability $1/2$. Under mechanism A, exactly one uncertain reference is

retained uniformly; under B, both or neither are retained with probability $1/2$. The two mechanisms have identical retention marginals and both satisfy $\Sigma_i \preceq 2D_i$. With exact retention likelihoods, $\kappa_i = 2$, $\epsilon_i = 0$, and $\beta_{i,n} = 1/(n+1)^2$.

Their expected retained-reference targets are

$$\bar{\mathbf{q}}^{\mathrm{f}}_{i,A} = \frac{n\mathbf{u} + \mathbf{v}}{n+1}, \qquad \bar{\mathbf{q}}^{\mathrm{f}}_{i,B} = \frac{(n+1)\mathbf{u} + \mathbf{v}}{n+2}.$$

Their separation satisfies

$$\begin{aligned}\left\|\bar{\mathbf{q}}^{\mathrm{f}}_{i,A} - \bar{\mathbf{q}}^{\mathrm{f}}_{i,B}\right\|_2 &= \frac{\|\mathbf{u} - \mathbf{v}\|_2}{(n+1)(n+2)} \\ &= \frac{n+1}{n+2}\frac{\beta_{i,n}}{\sqrt{2}}.\end{aligned}$$

Since the fixed reference quantities, retention marginals, and dependence bound are identical under A and B, any estimator $\widehat{\mathbf{q}}$ using only this information has the same output distribution under both mechanisms. Therefore,

$$\begin{aligned}\max_{m\in\{A,B\}} \mathbb{E}_{\mathrm{alg}}\left[\left\|\widehat{\mathbf{q}} - \bar{\mathbf{q}}^{\mathrm{f}}_{i,m}\right\|_2\right] &\geq \frac{1}{2}\left\|\bar{\mathbf{q}}^{\mathrm{f}}_{i,A} - \bar{\mathbf{q}}^{\mathrm{f}}_{i,B}\right\|_2 \\ &\geq \frac{\beta_{i,n}}{4\sqrt{2}}.\end{aligned}$$

Since $\beta_{i,n} \to 0$ as $n \to \infty$, no estimator using only this information can achieve uniformly $o(\beta_i)$ error. Hence the $O(\beta_i)$ rate is worst-case optimal. □

A.2 Proof of Theorem 4.4

Proof. For any sample i with strictly positive $\mathbf{q}^{\mathrm{L}}_i$ and $\mathbf{q}^{\mathrm{u}}_i$ and nonuniform probabilities across the remaining classes, omit the sample index. We hold the reference $\mathbf{q}^{\mathrm{L}}$ fixed, so the conditional reference probabilities u_c remain unchanged. For a given local change in the training label probability $\mathrm{d}q^{\mathrm{u}}_y$, we determine how the remaining class probabilities should change while preserving normalization and $H(\mathbf{q}^{\mathrm{u}}) = H(\mathbf{q}^{\mathrm{L}})$.

Step 1: Local minimum-distortion problem. For a local perturbation of $\mathbf{q}^{\mathrm{u}}$, the second-order term of $\mathrm{KL}(\cdot\|\mathbf{q}^{\mathrm{u}})$ is

$$\frac{1}{2}\sum_c \frac{(\mathrm{d}q^{\mathrm{u}}_c)^2}{q^{\mathrm{u}}_c}.$$

Normalization requires $\sum_c \mathrm{d}q^{\mathrm{u}}_c = 0$, while differentiating the entropy constraint gives $\sum_c \log q^{\mathrm{u}}_c \,\mathrm{d}q^{\mathrm{u}}_c = 0$. Since $\mathrm{d}q^{\mathrm{u}}_y$ is given, its contribution to the quadratic objective is fixed, and the remaining changes solve

$$\begin{aligned}\min_{\{\mathrm{d}q^{\mathrm{u}}_c\}_{c\neq y}} \quad & \frac{1}{2}\sum_{c\neq y}\frac{(\mathrm{d}q^{\mathrm{u}}_c)^2}{q^{\mathrm{u}}_c} \\ \text{subject to} \quad & \sum_{c\neq y}\mathrm{d}q^{\mathrm{u}}_c = -\mathrm{d}q^{\mathrm{u}}_y, \\ & \sum_{c\neq y}\log q^{\mathrm{u}}_c \,\mathrm{d}q^{\mathrm{u}}_c = -\log q^{\mathrm{u}}_y \,\mathrm{d}q^{\mathrm{u}}_y.\end{aligned}$$

Thus, the local problem determines the least KL-distorting adjustment of the remaining class probabilities for the given training label probability change while preserving normalization and entropy.

Step 2: Local change induced by the normalized power transformation. Differentiating Equation (10) with the reference probabilities u_c fixed gives, for $c \neq y$,

$$\frac{\mathrm{d}q^{\mathrm{u}}_c}{q^{\mathrm{u}}_c} = -\frac{\mathrm{d}q^{\mathrm{u}}_y}{1-q^{\mathrm{u}}_y} + \left(\log u_c - \sum_{k\neq y}\frac{q^{\mathrm{u}}_k}{1-q^{\mathrm{u}}_y}\log u_k\right)\mathrm{d}\alpha.$$

The first term adjusts the total probability mass assigned to the remaining classes, while the second changes their relative probabilities through α.

We next show that α can vary locally while preserving entropy. Holding q_y^{u} fixed and differentiating with respect to α gives

$$\frac{\partial H(\mathbf{q}^{\mathrm{u}})}{\partial \alpha} = -\alpha \sum_{c \neq y} q_c^{\mathrm{u}} \left(\log u_c - \sum_{k \neq y} \frac{q_k^{\mathrm{u}}}{1 - q_y^{\mathrm{u}}} \log u_k \right)^2 .$$

For the ECDO transformation, $\alpha > 0$, and nonuniformity across the remaining classes makes this derivative strictly negative. Hence, by the implicit function theorem, every sufficiently small change in q_y^{u} admits a unique smooth local adjustment of α that preserves $H(\mathbf{q}^{\mathrm{u}}) = H(\mathbf{q}^{\mathrm{L}})$. Normalization is already enforced by Equation (10), so the resulting change is feasible for the local problem in Step 1.

Step 3: Minimum-distortion property. The quadratic objective in Step 1 is strictly convex, and its gradient with respect to each remaining-class change is $\mathrm{d}q_c^{\mathrm{u}}/q_c^{\mathrm{u}}$. The difference between any two feasible adjustments has zero sum and zero inner product with $\log q_c^{\mathrm{u}}$ over classes $c \neq y$. Therefore, a feasible adjustment satisfies the first-order optimality condition when its relative changes are affine in $\log q_c^{\mathrm{u}}$.

From Equation (10),

$$\log q_c^{\mathrm{u}} = \log(1 - q_y^{\mathrm{u}}) + \alpha \log u_c - \log \sum_{k \neq y} u_k^{\alpha}, \qquad c \neq y.$$

Thus, $\log u_c$ is affine in $\log q_c^{\mathrm{u}}$ across the remaining classes. The relative changes derived in Step 2 are therefore affine in $\log q_c^{\mathrm{u}}$ and satisfy the first-order optimality condition. The strict convexity then implies that the adjustment induced by Equation (10) is the unique minimizer.

Finally, substituting the local change from Step 2 into the full quadratic term gives

$$\frac{1}{2} \sum_c \frac{(\mathrm{d}q_c^{\mathrm{u}})^2}{q_c^{\mathrm{u}}} = \frac{(\mathrm{d}q_y^{\mathrm{u}})^2}{2q_y^{\mathrm{u}}(1 - q_y^{\mathrm{u}})} + \frac{(\mathrm{d}\alpha)^2}{2} \sum_{c \neq y} q_c^{\mathrm{u}} \left(\log u_c - \sum_{k \neq y} \frac{q_k^{\mathrm{u}}}{1 - q_y^{\mathrm{u}}} \log u_k \right)^2 .$$

The cross term vanishes because the centered log probabilities have zero weighted sum. The first term is the unavoidable local distortion associated with changing the training label probability, while the second is the additional adjustment required to preserve entropy. Therefore, for a given local change in the training label probability, Equation (10) induces the unique entropy-preserving adjustment minimizing the second-order term of $\mathrm{KL}(\cdot \| \mathbf{q}^{\mathrm{u}})$. □

B Experimental Setup

In this section, we provide further details on our experimental setup. Appendix B.1 describes the construction of forgetting likelihoods, and Appendix B.2 details the training and optimization settings. All experiments are implemented in PyTorch (Paszke et al., 2019) and conducted on a single NVIDIA RTX PRO 6000 GPU. We repeat each experiment three times with different random seeds.

B.1 Forgetting Likelihood Construction

We consider settings where the true forget set is not known exactly and a detector instead provides sample-wise evidence for forgetting. For all detectors, we convert the resulting scores into continuous forgetting likelihoods using a Gaussian mixture model (GMM). This approach is motivated by noisy-label learning methods that use mixture posteriors to estimate whether training labels are correct (Arazo et al., 2019; Li et al., 2020). We define p_i as the posterior probability of the component associated with stronger forgetting evidence. This conversion does not use ground-truth forget set membership or the known corruption or poisoning rate. UADU directly uses the continuous likelihoods p_i. For the approximate baselines, we construct retain and forget sets by thresholding these likelihoods. Retrain uses the ground-truth retain set.

For the main mislabeled-data setting, we introduce 10% symmetric label noise and use the corrupted samples as the ground-truth forget set. We compute AUM scores during Original training following Pleiss et al. (2020). Lower AUM values indicate stronger evidence of mislabeling.

As an alternative source of forgetting evidence, we use EL2N (Paul et al., 2021) under the same 10% mislabeled-data setting. Following its early-training evaluation, we compute EL2N at epoch 10 and average the scores over three independently initialized auxiliary ResNet-18 models trained with the same schedule as Original. Larger EL2N values are treated as stronger forgetting evidence.

For the backdoor setting, we follow the sinusoidal poisoning setup of Barni et al. (2019) using the public implementation released by Qi et al. (2023). We poison 5% of the full training set while keeping the original labels unchanged. We use 5% rather than 10% because the poisoning is restricted to the target class. On CIFAR-10, a 10% overall poisoning rate would cover the entire target class. We use STRIP (Gao et al., 2019) to obtain sample-wise poisoning evidence by mixing each input with 100 clean images and computing the mean predictive entropy. We use the negative mean entropy as the score, so that larger values indicate stronger forgetting evidence.

B.2 Implementation Details

Original and baselines. We configure all baselines based on the hyperparameter settings reported in their original papers and publicly available implementations. Original and Retrain are trained for 100 epochs using SGD with an initial learning rate of 0.1 and cosine decay. FT, MUNBa, and SalUn use SGD for 10 epochs with a constant learning rate of 10^{-2}, whereas GA uses SGD for 5 epochs with a constant learning rate of 10^{-4}. SalUn uses a saliency-mask ratio of 0.5. For LAF+R, we use Adam with learning rates of 10^{-3}, 10^{-5}, 4×10^{-5}, and 5×10^{-5} for VAE training, feature-extractor optimization, alignment, and classifier repair, respectively. The VAE is trained for 10 epochs on CIFAR-10 and CIFAR-100 and 5 epochs on TinyImageNet, followed by 40 unlearning epochs and one repair epoch. AMUN uses SGD for 10 epochs with an initial learning rate of 10^{-2}, reduced by a factor of 0.1 after each epoch, and generates adversarial examples using 50-step ℓ_2 PGD.

UADU. We optimize LCUO and ECDO using Adam with cosine learning-rate decay. The initial learning rate is 7×10^{-4} for LCUO and 3×10^{-4} for ECDO. On CIFAR-10 and CIFAR-100, we use 200 LCUO steps and 600 ECDO steps. On TinyImageNet, we use 400 LCUO steps and $3{,}600$ ECDO steps.

C Additional Experiments

This section provides additional experiments examining the robustness of UADU beyond the main experimental setting. Our key analyses include:

- In Appendix C.1, we evaluate the robustness of UADU to changes in the forget set size by varying the label-noise rate from 5% to 15%.
- In Appendix C.2, we examine the sensitivity of LCUO and ECDO to their learning rates and numbers of optimization steps.
- In Appendix C.3, we study the effect of the threshold used by conventional baselines to convert forgetting likelihoods into explicit retain and forget sets.
- In Appendix C.4, we extend the evaluation to VGG16-BN to examine the robustness of UADU to a change in model architecture.

C.1 Robustness to Forget Set Size

We evaluate robustness to the forget set size by varying the symmetric label-noise rate over 5%, 10%, and 15%. In our mislabeled-data setting, the mislabeled samples define the ground-truth forget set, so the noise rate directly controls its size. For each setting, we train the corresponding Original and Retrain models and recompute the AUM-based forgetting likelihoods, while keeping the unlearning hyperparameters fixed.

As shown in Table 3, UADU achieves the A_f closest to Retrain at all three forget set sizes. This behavior remains consistent as the forget set size increases, showing that the forgetting performance

of UADU is not tied to a particular forget set size. UADU also yields the MIA closest to Retrain at 5% and 10%, and the second closest at 15%. Overall, these results show that UADU maintains retraining-like forgetting behavior across different forget set sizes without changing its unlearning hyperparameters.

Table 3: Sensitivity to forget set size on CIFAR-10. Results are averaged over three random seeds. For A_r, A_f, and MIA, results closer to Retrain are better. **Bold** indicates the best result and underlining indicates the second best result among unlearning methods.

Method	Forget Set 5%			Forget Set 10%			Forget Set 15%		
	A_r (%)	A_f (%)	MIA (%)	A_r (%)	A_f (%)	MIA (%)	A_r (%)	A_f (%)	MIA (%)
Original	99.98$_{\pm 0.01}$	99.59$_{\pm 0.06}$	55.09$_{\pm 0.09}$	99.95$_{\pm 0.01}$	99.25$_{\pm 0.10}$	66.81$_{\pm 0.74}$	99.90$_{\pm 0.04}$	99.06$_{\pm 0.11}$	71.13$_{\pm 2.12}$
Retrain	100.00$_{\pm 0.00}$	0.56$_{\pm 0.16}$	76.13$_{\pm 0.53}$	100.00$_{\pm 0.00}$	0.67$_{\pm 0.06}$	75.75$_{\pm 0.48}$	99.83$_{\pm 0.09}$	0.87$_{\pm 0.16}$	79.41$_{\pm 0.68}$
FT	98.99$_{\pm 1.67}$	43.76$_{\pm 32.50}$	27.40$_{\pm 30.24}$	99.82$_{\pm 0.08}$	38.24$_{\pm 2.13}$	14.07$_{\pm 1.99}$	99.56$_{\pm 0.20}$	31.88$_{\pm 8.55}$	25.23$_{\pm 10.14}$
GA	**99.99**$_{\pm 0.01}$	86.40$_{\pm 8.85}$	32.68$_{\pm 3.37}$	**99.89**$_{\pm 0.09}$	43.25$_{\pm 5.19}$	43.18$_{\pm 5.77}$	98.96$_{\pm 0.97}$	28.00$_{\pm 6.66}$	**56.87**$_{\pm 5.70}$
MUNBa	98.54$_{\pm 2.24}$	26.36$_{\pm 15.51}$	26.17$_{\pm 26.25}$	99.85$_{\pm 0.03}$	31.05$_{\pm 1.39}$	14.91$_{\pm 0.95}$	**99.76**$_{\pm 0.07}$	26.78$_{\pm 2.22}$	18.00$_{\pm 6.16}$
LAF+R	96.02$_{\pm 0.65}$	11.04$_{\pm 1.08}$	7.92$_{\pm 1.42}$	97.23$_{\pm 0.65}$	13.42$_{\pm 0.90}$	9.13$_{\pm 0.70}$	96.61$_{\pm 1.55}$	12.20$_{\pm 5.08}$	19.60$_{\pm 16.02}$
SalUn	98.60$_{\pm 0.38}$	11.48$_{\pm 3.88}$	33.03$_{\pm 10.20}$	99.26$_{\pm 0.09}$	12.82$_{\pm 1.43}$	34.87$_{\pm 4.65}$	99.16$_{\pm 0.10}$	12.17$_{\pm 0.91}$	37.30$_{\pm 3.33}$
AMUN	98.66$_{\pm 0.17}$	59.45$_{\pm 3.48}$	11.89$_{\pm 0.50}$	98.91$_{\pm 0.04}$	55.91$_{\pm 1.11}$	14.90$_{\pm 0.42}$	98.45$_{\pm 0.14}$	55.56$_{\pm 2.28}$	19.87$_{\pm 2.36}$
UADU (ours)	97.80$_{\pm 0.31}$	**4.12**$_{\pm 0.37}$	**52.40**$_{\pm 3.07}$	98.55$_{\pm 0.13}$	**5.93**$_{\pm 0.47}$	**51.01**$_{\pm 2.49}$	98.41$_{\pm 0.23}$	**6.22**$_{\pm 0.82}$	49.85$_{\pm 3.77}$

C.2 Robustness to Optimization Hyperparameters

We evaluate the robustness of UADU to the learning rate and number of optimization steps in each stage on CIFAR-10. For LCUO and ECDO separately, we vary one optimization hyperparameter around its default value while keeping all other settings fixed.

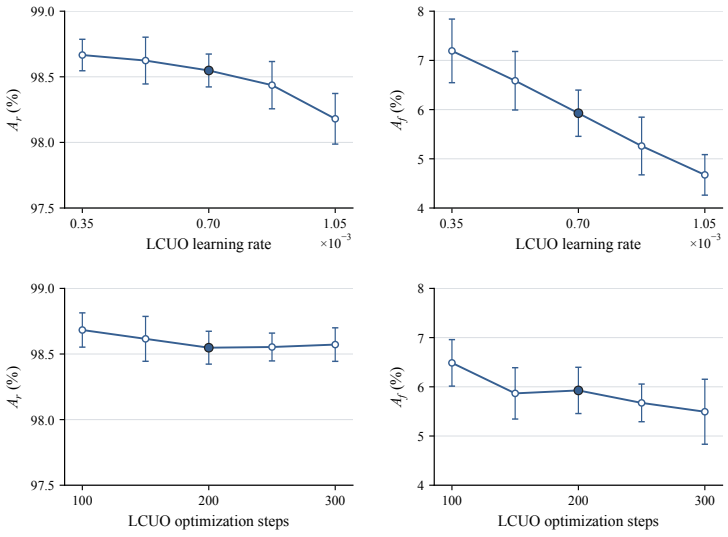


Figure 4: Sensitivity to LCUO optimization hyperparameters on CIFAR-10. We vary the learning rate and number of optimization steps while keeping all other settings fixed, and report retain accuracy (A_r) and forget accuracy (A_f). Error bars indicate the standard deviation over three random seeds, and filled markers denote the default configuration used in the main experiments.

As shown in Figures 4 and 5, UADU remains robust across the tested optimization hyperparameters, with the two components exhibiting different sensitivity patterns. For LCUO, increasing the learning rate progressively strengthens forgetting while A_r remains relatively stable, whereas varying the number of optimization steps has only a limited effect on both metrics. ECDO is less sensitive to its learning rate, and increasing the number of optimization steps mainly improves A_r at the lower end before both metrics become stable. These results show that similar retain and forget behavior is maintained over a range of learning rates and optimization steps. The performance also changes

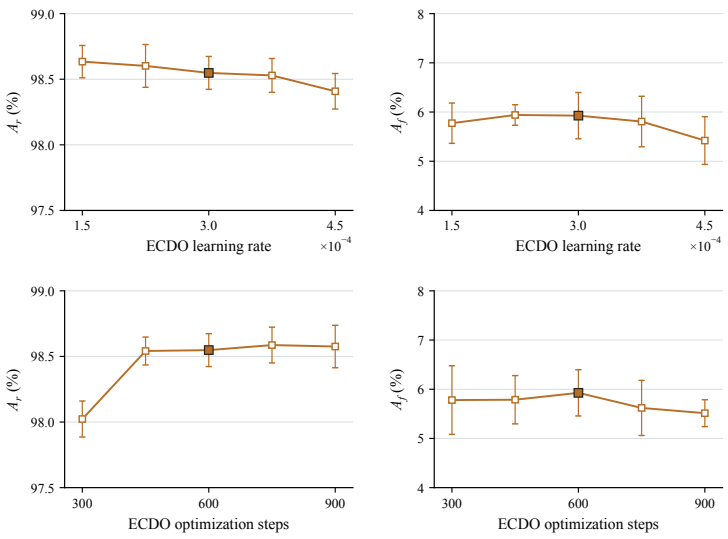


Figure 5: Sensitivity to ECDO optimization hyperparameters on CIFAR-10. We vary the learning rate and number of optimization steps while keeping all other settings fixed, and report retain accuracy (A_r) and forget accuracy (A_f). Error bars indicate the standard deviation over three random seeds, and filled markers denote the default configuration used in the main experiments.

smoothly around the default settings, without a large drop in either metric. Overall, UADU shows robust unlearning behavior across the optimization settings.

C.3 Robustness to Forget Set Threshold

We examine the effect of converting forgetting likelihoods into explicit retain and forget sets. For the conventional baselines, we construct the estimated forget set as $\widehat{\mathcal{D}}_f(\tau) = \{(x_i, y_i) : p_i \geq \tau\}$ and vary $\tau \in \{0.3, 0.5, 0.7\}$ while keeping all other experimental settings fixed. UADU does not construct such a partition and instead uses the continuous forgetting likelihoods directly. Its result is therefore unchanged across these threshold settings.

As shown in Table 4, the performance of the set-based baselines changes with the threshold used to construct the forget set. In contrast, UADU is unaffected by this choice and remains closest to Retrain in both A_f and MIA across all three threshold settings. These results show that UADU avoids the sensitivity introduced by converting uncertain forgetting evidence into a hard retain–forget partition.

Table 4: Robustness to the forget set threshold on CIFAR-10 with 10% label noise. Results are averaged over three random seeds. For A_r, A_f, and MIA, results closer to Retrain are better. **Bold** indicates the best result and underlining indicates the second best result among unlearning methods.

Method	$\tau = 0.3$			$\tau = 0.5$			$\tau = 0.7$		
	A_r (%)	A_f (%)	MIA (%)	A_r (%)	A_f (%)	MIA (%)	A_r (%)	A_f (%)	MIA (%)
Original	$99.95_{\pm 0.01}$	$99.25_{\pm 0.10}$	$66.81_{\pm 0.74}$	$99.95_{\pm 0.01}$	$99.25_{\pm 0.10}$	$66.81_{\pm 0.74}$	$99.95_{\pm 0.01}$	$99.25_{\pm 0.10}$	$66.81_{\pm 0.74}$
Retrain	$100.00_{\pm 0.00}$	$0.67_{\pm 0.06}$	$75.75_{\pm 0.48}$	$100.00_{\pm 0.00}$	$0.67_{\pm 0.06}$	$75.75_{\pm 0.48}$	$100.00_{\pm 0.00}$	$0.67_{\pm 0.06}$	$75.75_{\pm 0.48}$
FT	$\underline{99.86}_{\pm 0.01}$	$40.48_{\pm 1.72}$	$12.94_{\pm 0.14}$	$99.82_{\pm 0.08}$	$38.24_{\pm 2.13}$	$14.07_{\pm 1.99}$	$99.61_{\pm 0.39}$	$30.05_{\pm 17.70}$	$29.24_{\pm 24.95}$
GA	$\mathbf{99.92}_{\pm 0.06}$	$48.64_{\pm 6.18}$	$\underline{38.87}_{\pm 5.77}$	$\mathbf{99.89}_{\pm 0.09}$	$43.25_{\pm 5.19}$	$\underline{43.18}_{\pm 5.77}$	$\mathbf{99.88}_{\pm 0.13}$	$43.20_{\pm 7.79}$	$\underline{43.15}_{\pm 8.88}$
MUNBa	$99.84_{\pm 0.01}$	$29.44_{\pm 0.67}$	$13.83_{\pm 1.00}$	$\underline{99.85}_{\pm 0.03}$	$31.05_{\pm 1.39}$	$14.91_{\pm 0.95}$	$\underline{99.75}_{\pm 0.25}$	$26.98_{\pm 12.01}$	$20.84_{\pm 16.24}$
LAF+R	$97.04_{\pm 0.72}$	$12.38_{\pm 0.76}$	$10.84_{\pm 3.63}$	$97.23_{\pm 0.65}$	$13.42_{\pm 0.90}$	$9.13_{\pm 0.70}$	$97.71_{\pm 0.31}$	$14.60_{\pm 1.78}$	$8.85_{\pm 0.18}$
SalUn	$99.08_{\pm 0.08}$	$\underline{12.34}_{\pm 1.46}$	$35.36_{\pm 4.77}$	$99.26_{\pm 0.09}$	$\underline{12.82}_{\pm 1.43}$	$34.87_{\pm 4.65}$	$99.35_{\pm 0.14}$	$\underline{13.31}_{\pm 2.78}$	$35.02_{\pm 9.01}$
AMUN	$98.59_{\pm 0.12}$	$56.57_{\pm 1.00}$	$15.51_{\pm 0.61}$	$98.91_{\pm 0.04}$	$55.91_{\pm 1.11}$	$14.90_{\pm 0.42}$	$99.01_{\pm 0.08}$	$55.39_{\pm 0.89}$	$15.53_{\pm 0.19}$
UADU (ours)	$98.55_{\pm 0.13}$	$\mathbf{5.93}_{\pm 0.47}$	$\mathbf{51.01}_{\pm 2.49}$	$98.55_{\pm 0.13}$	$\mathbf{5.93}_{\pm 0.47}$	$\mathbf{51.01}_{\pm 2.49}$	$98.55_{\pm 0.13}$	$\mathbf{5.93}_{\pm 0.47}$	$\mathbf{51.01}_{\pm 2.49}$

C.4 Robustness to Model Architecture

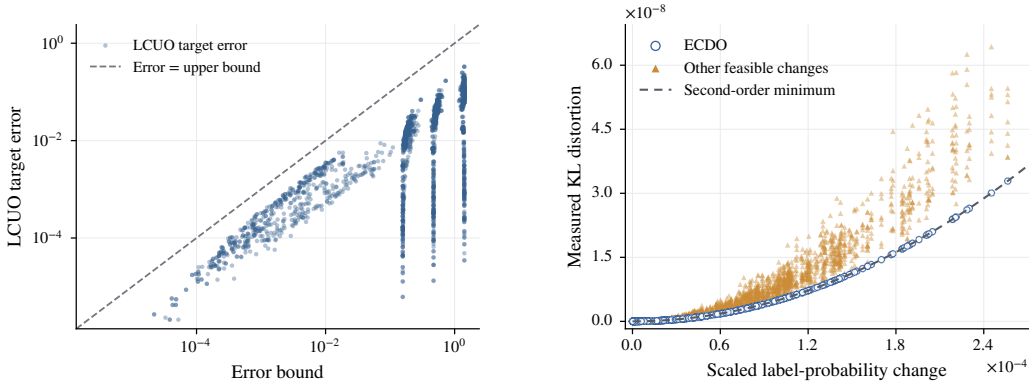
We further evaluate UADU with VGG16-BN on CIFAR-10 to examine whether its forgetting behavior depends on the ResNet-18 architecture used in the main experiments. We use the same 10% mislabeled-data setting and retain the LCUO and ECDO learning rates used for ResNet-18, without architecture-specific learning-rate tuning.

Table 5: Robustness to model architecture on CIFAR-10 using VGG16-BN with 10% label noise. Results are averaged over three random seeds. For A_r, A_f, and MIA, results closer to Retrain are better. **Bold** indicates the best result and underlining indicates the second best result among unlearning methods.

Method	A_r (%)	A_f (%)	MIA (%)
Original	$99.94_{\pm 0.01}$	$99.02_{\pm 0.20}$	$64.77_{\pm 2.68}$
Retrain	$100.00_{\pm 0.00}$	$0.86_{\pm 0.10}$	$76.83_{\pm 0.69}$
FT	$\mathbf{98.81}_{\pm 0.27}$	$18.99_{\pm 1.48}$	$\underline{53.77}_{\pm 3.02}$
GA	$97.30_{\pm 2.57}$	$32.73_{\pm 5.51}$	$\mathbf{64.39}_{\pm 4.69}$
MUNBa	$\underline{98.71}_{\pm 0.09}$	$23.63_{\pm 0.48}$	$47.07_{\pm 3.01}$
LAF+R	$94.53_{\pm 0.77}$	$\underline{11.49}_{\pm 0.15}$	$28.68_{\pm 8.58}$
SalUn	$98.48_{\pm 0.21}$	$12.87_{\pm 1.40}$	$47.80_{\pm 1.28}$
AMUN	$97.72_{\pm 0.15}$	$56.57_{\pm 1.84}$	$11.96_{\pm 0.23}$
UADU (ours)	$96.94_{\pm 0.13}$	$\mathbf{4.45}_{\pm 0.30}$	$50.73_{\pm 0.56}$

As shown in Table 5, UADU again achieves the A_f closest to Retrain after replacing ResNet-18 with VGG16-BN. Its absolute deviation from Retrain in A_f is 3.59 percentage points, compared with 10.63 percentage points for the next-closest unlearning method. Although some baselines are closer to Retrain in A_r or MIA, the forgetting advantage of UADU is preserved under the architecture change. This result shows that its likelihood-conditioned forgetting mechanism is not specific to the ResNet-18 architecture or to architecture-specific learning rate tuning.

D Theoretical Analysis



(a) LCUO target-error bound.

(b) ECDO local KL optimality.

Figure 6: Empirical analysis of the theoretical properties of LCUO and ECDO. (a) Comparison of the LCUO target error with the theoretical upper bound in Theorem 4.3. (b) Comparison of the KL distortion induced by ECDO and alternative feasible redistributions with the second-order minimum in Theorem 4.4.

We empirically examine the two theorems underlying the design of UADU. For LCUO, we evaluate the approximation error of the reference-based forget target under uncertain retention evidence. For

ECDO, we evaluate the local second-order KL distortion induced when adjusting the prescribed training label probability while preserving entropy. These properties correspond to Theorems 4.3 and 4.4, respectively.

Figure 6a compares the observed LCUO forget-target error with the upper bound derived in Theorem 4.3. Specifically, the horizontal axis shows the theoretical bound $\sqrt{2}\epsilon_i + 2\sqrt{2}\beta_i$, while the vertical axis shows the corresponding target error. The observed errors remain below the equality line across the evaluated cases, empirically supporting the error-control property of the reference-based target construction.

Figure 6b examines the local minimum-distortion property of ECDO. We perturb the prescribed training label probability within the local regime considered in Theorem 4.4 and compare the resulting KL distortion with the derived second-order minimum and alternative feasible redistributions. The distortion induced by ECDO closely follows the second-order minimum, whereas the alternative feasible changes generally incur larger distortion. This result empirically supports the local second-order KL optimality of the redistribution.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing and translation, primarily to improve the clarity, readability, and overall presentation of author-developed content. We did not use generative AI tools for theoretical or conceptual development, mathematical claims or proofs, hypothesis development, research methodology or experimental design, method implementation, or interpretation of experimental results. Other required-disclosure tasks, including synthetic-data generation, dataset cleaning or reformatting, and qualitative or thematic data analysis, were not applicable to this work.

Additionally, we used generative AI tools for minor code editing and debugging and for searching for relevant information and literature. All AI-assisted code modifications were reviewed and tested by the authors, and identified information and references were verified against the original sources. We reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.