The Role of Learning and Memorization in Relabeling-Based Unlearning

Anonymous authors
Paper under double-blind review

Abstract

This work studies how the nature of a response generated by a model impacts the efficiency of *relabeling-based unlearning*, a common unlearning technique that trains the model to fit an *unlearn* set (i.e., a dataset that we wish the model to unlearn) with alternative responses to prevent it from generating unwanted outputs that align with the unlearn set. We distinguish between two different ways models can generate undesirable outputs: **learning-based generation**, where the model learns an underlying *rule* connecting the input and the response (e.g., social stereotypes), and **memorization-based generation**, where the model memorizes specific information about a given input (e.g., private information like a phone number). We demonstrate through binary classification and TOFU-LM experiments that relabeling-based unlearning converges more slowly and causes greater temporary retain-accuracy loss for learning-based generation than for memorization-based generation.

1 Introduction

Deep learning models have shown remarkable capability and led to many successful applications driven by increasingly larger architectures and datasets. However, their training process may also result in undesirable model behaviors with serious safety risks, such as privacy leakage, social bias, and creation of harmful content (Carlini et al., 2021; Huang et al., 2022; Sandbrink, 2023). To address these risks, machine unlearning has emerged as an increasingly growing field aimed at removing the influence of undesirable data from a model without full retraining. This capability is especially critical for foundation models like Large Language Models (LLMs), where the massive scale of data and computation cost makes retraining prohibitively expensive or even infeasible.

A primary objective of machine unlearning is to prevent the model from producing undesirable outputs, represented by a data set known as the *unlearn* set (also known as the forget set), given relevant inputs without compromising the model's overall capability. Researchers have proposed various unlearning methods based on different frameworks (Jang et al., 2022; Zhang et al., 2024; Jiang et al., 2020; Li et al., 2024). Among these methods, a common approach is the *relabeling-based method* (Deeb and Roger, 2024; Maini et al., 2024; Eldan and Russinovich, 2023). The core idea for the relabeling-based method is to first create a new unlearn set with the same inputs as the original unlearn set but with alternative, harmless responses. The model is then trained with this new set being a part of the training data. By training the model to fit these new responses, we effectively encourage the model to predict the original inputs with alternative responses and then overwrite the undesirable responses.

While the field of machine unlearning is receiving increasing attention, the understanding on what factors truly influence its difficulty and efficiency is relatively underexplored. This is a crucial area of study since the understanding of these factors allows us to gain deeper insights on how these unlearning methods work and help us develop more effective and reliable unlearning methods. Prior works in this area have identified several key factors that influence the unlearning efficiency including frequency of unlearn data in the training data (Krishnan et al., 2025), data entanglement and memorization level (Zhao et al., 2024a), robustness to parameter perturbation (Feng et al., 2025), how knowledge is encoded in the

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.

training data (Wu et al., 2025) and so on. We defer a detailed discussion of related work to Appendix A.

Following this line of work, we study a new factor affecting the difficulty and efficiency of *relabeling-based unlearning*: how the model generates undesirable responses after initial training. We distinguish between **learning-based generation**, which applies a general rule connecting inputs to responses (e.g., a genre-to-recognition stereotype), and **memorization-based generation**, which recalls input-specific responses (e.g., a personal phone number). This mechanism-based distinction differs from the example-level memorization measure of Zhao et al. (2024a), which quantifies a prediction's dependence on an individual training example. Whereas their main evaluation measures accuracy agreement with retraining, we focus on unlearning efficiency and convergence speed while preserving retain accuracy.

In this work, we study how the nature of response generation (learning-based versus memorization-based) affects the unlearning efficiency for the relabeling-based method. We develop a theoretical framework to model the unlearning process, analyze model belief changes, and establish lower bound results demonstrating the need for hypothesis space expansion for learning-based generation.

The key contributions of this work are highlighted as follows:

- We propose a hypothesis-testing framework for relabeling-based unlearning with mathematical models of learning- and memorization-based hypotheses. Relabeling is modeled as a sequence of *"fabricated evidence"* that contradicts predictions based on previous observations.
- We analyze how conflicting evidence changes beliefs in these hypotheses relative to baselines that make constant predictions. We show that the memorization hypothesis is more stable: its log-belief drop is bounded independently of evidence length, while the learning hypothesis's belief drop grows with evidence length.
- We establish a lower bound showing that fitting even a single relabeled point can cause substantial performance loss within a hypothesis class that perfectly fits the original task distribution. This suggests that fitting the relabeled data while preserving overall performance may require revising the model's prior and expanding its hypothesis space to include more complex hypotheses, potentially leading to slower convergence and worse performance.
- We evaluate binary classification and TOFU-LM question answering, which contrasts a shared genre-to-recognition stereotype with author-specific assignments. Under Standard Relabeling and AltPO, memorization-based generation converges faster with smaller retain-accuracy drops and faster recovery. This gap extends to held-out paraphrases, and the binary discrepancy persists across forget-set sizes. Experiments with NPO (Zhang et al., 2024) and RMU (Li et al., 2024) suggest that the gap extends beyond relabeling, depending on the method.

2 Preliminaries

2.1 Machine Unlearning

Machine unlearning aims to remove the influence of an undesired *unlearn dataset* (U) from a model θ, while preserving overall utility using a *retain dataset* (R) (Jang et al., 2022; Zhang et al., 2024; Jiang et al., 2020; Li et al., 2024).

Most existing machine unlearning methods are achieved by finetuning model θ over an objective written as

$$\min_{\theta}(1-\alpha)L_U(\theta, U) + \alpha L_R(\theta, R)$$

where L_U measures unlearning effectiveness on U, L_R is the retain loss for preserving model utility on R, and α balances the two objectives.

2.2 Relabeling-based Machine Unlearning

Among the various unlearning methods, an important approach is the relabeling-based method, which has been explored in recent work (Yu et al., 2023; Yao et al., 2024; Eldan and Russinovich, 2023; Ishibashi and Shimodaira, 2023; Maini et al., 2024; Deeb and Roger, 2024; Fan et al., 2023). Relabeling-based unlearning first constructs a new unlearn set U' by replacing each input-label pair (x, y) in U with (x, y'), where $y' \neq y$. The alternative response can be deliberately crafted, such as "I don't know" (Maini et al., 2024), or a randomly selected but sensible label, such as an incorrect choice for a multiple-choice question (Deeb and Roger, 2024).

After the construction of U', the overall unlearning objective is

$$\min_{\theta}(1-\alpha)L(\theta, U') + \alpha L(\theta, R) \tag{1}$$

where L may be a standard prediction loss that directly fits the alternative responses (Deeb and Roger, 2024), or a preference optimization loss that favors these alternatives over the original responses (Mekala et al., 2025). The logic behind relabeling-based unlearning is that by training the model on these relabeled pairs (U'), we encourage the model to predict the original inputs with alternating or neutral labels, effectively overwriting the undesired information.

3 Model's Beliefs Update for Unlearning via Relabeling

In this section, we model relabeling-based unlearning as presenting conflicting evidence, termed *fabricated evidence*, and analyze how it updates beliefs in learning- and memorization-based hypotheses.

A learning-based hypothesis applies a general rule to generate a response. For example, given the question *"What is the solution to* $3x + 5 = 11$*?"*, the model answers *"*$x = 2$*"* by applying arithmetic rules. In contrast, a memorization-based hypothesis memorizes a specific input-output mapping. Given *"What is the filename of Alice's medical record?"*, the model answers *"183"* based on a previously observed fact, rather than a general rule for deriving filenames.

Relabeling (Equation 1) trains the model on these same questions with alternative responses. Pairing the arithmetic question with *"*$x = 3$*"* provides evidence that contradicts the learned arithmetic rule. Similarly, pairing the medical-record question with *"154"* contradicts the memorized association between Alice's record and its filename. These contradictory examples prompt the model to update its beliefs about the hypotheses underlying its original predictions.

We show that the learning-based hypothesis is less stable under fabricated evidence. For repeated evidence at one input, its belief relative to a baseline hypothesis decreases exponentially with evidence length, whereas the memorization-based hypothesis's log-belief drop is bounded independently of evidence length when prior observations are limited. Revising learning-based predictions also requires finding a new hypothesis that fits both the relabeled unlearn data and retain data. This search may be slow when it requires substantial changes to the model's prior over the hypothesis space. In contrast, the memorization-based hypothesis permits more flexible and efficient prediction updates.

3.1 Belief Update Modeling

Given a sequence of input-output pairs $D = \{(x_j, z_j)\}_{j=1}^{n}$, we simplify our analysis by letting $x_j \in [N] = \{1, 2, \dots N\}$ and z_j is either -1 or $+1$ for any $j \in [n]$. Assume that the input x is uniformly sampled from $[N]$ and the observed response $z \in \{-1, +1\}$ is a noisy version of the true, underlying response $y \in \{-1, +1\}$. Specifically, the true response y is flipped with a probability $\epsilon \in (0, 0.5)$ independently for each observation.

We propose two primary hypotheses to distinguish between how the model predicts the response y given input x, a learning-based hypothesis (H_0) that the model learns the

underlying, general rule between x and y, and a memorization-based hypothesis (H_1) that the model memorizes every pair it has ever seen. The mathematical representations of both hypotheses are defined as follows:

- H_0 (Learning-based Hypothesis): The response y is determined by a known function $f : [N] \to \{-1, +1\}$, that is, $y = f(x)$.
- H_1 (Memorization-based Hypothesis): A latent vector $V \in \{-1, +1\}^N$ is sampled once where each element in V is i.i.d uniformly sampled from $\{-1, +1\}$. The response y for input x is the value stored at the x_{th} position of vector V, i.e., $y = V_x$.

Under the learning-based hypothesis (H_0), predictions follow a general rule f learned during initial training. Under the memorization-based hypothesis (H_1), input-output associations are determined by independently sampled latent labels V. Observations at one input therefore provide no information about labels at other inputs, and predictions rely on observations specific to each input.

The hypotheses also differ in their uncertainty at unobserved inputs. H_1 assumes independence between inputs, leading to high uncertainty on unobserved examples. Meanwhile, H_0 generalizes rules across inputs, resulting in low uncertainty for predictions on unseen data. This difference characterizes the two generation mechanisms in our theoretical model.

We compare the prior-posterior belief change of a hypothesis relative to baseline hypotheses. We define two baseline hypotheses that make constant predictions:

$$H_2 : y = +1 \text{ for all } x \in [N]; \quad H_3 : y = -1 \text{ for all } x \in [N];$$

Notation and belief change. Let h denote a history of n observations and e a sequence of k new observations. We write $P(H \mid h)$ and $P(H \mid h, e)$ for the beliefs in hypothesis H before and after incorporating e, and $P(y \mid H, h, x)$ for its predictive probability at input x.

We measure belief change by the decrease in log-posterior odds relative to a baseline hypothesis H_i, where $i \in \{2, 3\}$:

$$\Delta_e(H, H_i) = \log \frac{P(H \mid h)}{P(H_i \mid h)} - \log \frac{P(H \mid h, e)}{P(H_i \mid h, e)}. \tag{2}$$

We assume that the compared hypotheses have positive probability conditional on h. A positive $\Delta_e(H, H_i)$ indicates a decrease in belief in H relative to H_i.

We also define fabricated evidence against a hypothesis as a sequence of data points that consistently contradict the hypothesis's predictions, given its past observations.

Definition 1. (Fabricated Evidence) Given a hypothesis H with history observations h, we call a sequence of datapoints $e = \{(x_j, z_j)\}_{j=1}^k$ fabricated evidence against H, if we have

$$y_j = - \operatorname*{arg\,max}_{y \in \{-1,+1\}} P(y|H, h, x_j) \quad \forall j \in [k]$$

and z_j is a noisy observation of y_j with i.i.d flipping noise ϵ.

We consider fabricated evidence concentrated at a single input x^*, so that $x_j = x^*$ for all $j \in [k]$. The following two theorems characterize the resulting belief changes for learning- and memorization-based hypotheses. Their proofs appear in Appendix B.

Theorem 2 (Stability of Learning-based Hypothesis)**.** *Fix an observation history h and an input $x^* \in [N]$. Let $e = \{(x^*, z_j)\}_{j=1}^k$ be fabricated evidence against H_0 given h, and let H_i, $i \in \{2, 3\}$, be the constant-prediction baseline with label $-f(x^*)$. Let $l = |\{j : z_j = f(x^*)\}|$ count the noise flips in e. Then $l \sim \text{Binomial}(k, \epsilon)$ and*

$$\Delta_e(H_0, H_i) = (k - 2l) \log \left(\frac{1 - \epsilon}{\epsilon} \right).$$

For any fixed $\epsilon \in (0, 1/2)$,

$$\Delta_e(H_0, H_i) = \Omega(k)$$

with probability at least $1 - O(k^{-10})$ over the observation noise.

Theorem 3 (Stability of Memorization-based Hypothesis)**.** *Fix an observation history h and an input $x^* \in [N]$. Let $e = \{(x^*, z_j)\}_{j=1}^{k}$ be fabricated evidence against H_1 given h. For $z \in \{-1, +1\}$, let m_{z,x^*} denote the number of observations of (x^*, z) in h. Then, for every $i \in \{2, 3\}$ and every realization of e,*

$$\Delta_e(H_1, H_i) \leq \log\left(1 + \left(\frac{1-\epsilon}{\epsilon}\right)^{|m_{1,x^*} - m_{-1,x^*}|}\right).$$

In particular, if x^* does not appear in h, then

$$\Delta_e(H_1, H_i) \leq \log 2.$$

Remark. Theorems 2 and 3 show that the learning-based hypothesis is less stable under conflicting evidence. For H_0, the decrease in model belief grows linearly with evidence length with high probability, regardless of whether x^* appeared in the initial history. For H_1, the decrease is bounded independently of evidence length for any fixed history. Its bound increases with the historical label imbalance at x^*, and is at most $\log 2$ when x^* was previously unobserved. Therefore, sufficiently long conflicting evidence can make the baseline hypothesis dominate the learning-based hypothesis, while the memorization-based hypothesis's relative belief change remains bounded regardless of evidence length.

Hypothesis mixture: The hypotheses H_0 and H_1 isolate two generation mechanisms, but realistic predictions may combine both. For example, an arithmetic answer may be supported by a learned rule and a memorized solution to a previously encountered question. Our framework accommodates this by maintaining beliefs over a set $\mathcal{H}$ of hypotheses, including those defined above. The predictive distribution is

$$P(y \mid x, h) = \sum_{H \in \mathcal{H}} P(H \mid h) P(y \mid H, h, x).$$

Depending on the hypothesis predictions and their weights, generation may be leaning towards learning-based, memorization-based, or intermediate. Theorems 2 and 3 still describe how belief in each hypothesis changes relative to the baselines, even when the model uses both learning and memorization. The final prediction depends on what each hypothesis predicts and how much weight the model gives it.

3.2 Prediction Update for Relabeling-based Unlearning

Theorem 2 describes how conflicting evidence weakens support for the original shared rule f. The model may instead adopt another rule, but that replacement must fit the relabeled unlearn set U' while preserving predictions on the retain set R. A learning-based hypothesis represented by a simple rule may lack the flexibility to meet both requirements. This may require expanding the current hypothesis space to include more complex hypotheses, potentially slowing convergence.

Theorem 4 illustrates this limitation: even when a linear classifier perfectly fits the original task, fitting a single relabeled point can force constant error within the same hypothesis class. Preserving low error therefore requires a more expressive hypothesis class. The proof appears in Appendix B.

Theorem 4. *Given a d-dimensional linear hypothesis class defined as $\mathcal{H} = \{h_{w,b}(x) = sign(w^T x + b) | w \in \mathbb{R}^d, b \in \mathbb{R}\}$. There exists a distribution $\mathcal{D}$ and $U' = \{(x', y') \in [N]^d \times \{-1, +1\}\}$ where with $N \geq 3$, such that $\min_{h \in \mathcal{H}} err_D(h) = 0$ and $P_{\mathcal{D}}(x = x')$ is negligible, however, for any $\hat{h} \in \mathcal{H}$ with $\hat{h}(x') = y'$, we have*

$$err_D(\hat{h}) \geq 0.1$$

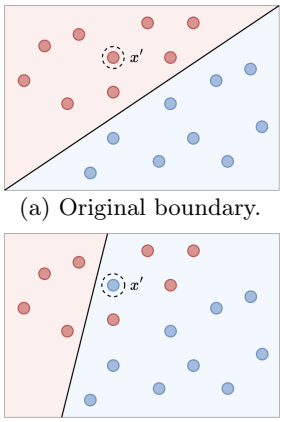
where $err_{\mathcal{D}}(\cdot)$ is the 0-1 error evaluated on distribution $\mathcal{D}$.

In the constructed task, every linear classifier that fits the relabeled point incurs constant population error. Achieving both accurate relabeling and low population error therefore requires a predictor outside the original linear class. This representational limitation helps explain for the slower convergence and retain-accuracy loss observed in our experiments.

In contrast, the memorization hypothesis (H_1) offers greater flexibility and can accommodate relabeling within its existing hypothesis space. Its maximum-likelihood prediction is

$$\hat{y}_x = \begin{cases} +1 & \text{if } m_{1,x} \geq m_{-1,x}, \\ -1 & \text{otherwise,} \end{cases}$$

where $m_{z,x}$ counts observations of label z at input x. Repeated observations of $(x, -1)$ reverse a positive prediction once $m_{-1,x} > m_{1,x}$, without changing predictions at other inputs under H_1. This local flexibility helps explain the faster and more stable relabeling-based unlearning observed in our experiments.



(a) Original boundary.

(b) After relabeling x'.

Figure 1: Illustration of Theorem 4. A linear boundary that fits the relabeled point x' misclassifies retained points.

4 Experiments

We test whether relabeling-based unlearning is slower and less stable for learning-based than memorization-based generation in controlled binary classification and TOFU-LM question answering. We also test whether the gap extends to non-relabeling methods.

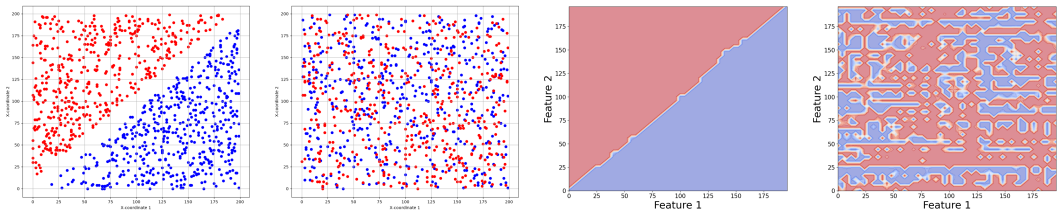
Experiment setting. For each dataset, we first fine-tune Llama-3.2-3B-Instruct on the full training set $R \cup U$ with original answers, starting from the same pretrained checkpoint and using answer-only cross-entropy. Each unlearning method then starts from the resulting fine-tuned checkpoint. Held-out questions are excluded from both stages. Our primary methods are Standard Relabeling, which fits alternative answers with prediction loss, and AltPO (Mekala et al., 2025), which prefers these alternatives over the original answers. All methods include retain-data training. We report means over three runs. Hyperparameters and additional-model results for Qwen3-4B (QwenTeam, 2025) and an 8B Llama model (Dubey et al., 2024) appear in Appendix C.

4.1 Binary Classification

Following Zhao et al. (2024b) and Dinh et al. (2022), we formulate binary classification as a language-model prediction task. Two-dimensional inputs allow us to control the labeling rule and visualize the resulting decision boundaries.

Tasks and data. We sample 1,000 distinct integer inputs $x = (x_1, x_2)$ in $[0, 200]^2$ with balanced labels. In LINEAR, $y = +1$ when $x_1 \geq x_2$ and $y = -1$ otherwise. In RANDOM, the labels are generated uniformly and independent of inputs and fixed throughout training. Thus, LINEAR admits a shared rule, whereas RANDOM requires input-specific associations. We verbalize -1 and $+1$ as `Foo` and `Bar`. For example, the LINEAR training prompt "*What is the label for this input? Input: 62 87. Label:*" has response `Foo`. We designate 30 inputs as the forget set and retain the remaining inputs. Standard Relabeling flips the forget labels, while AltPO uses the flipped and original labels as preferred and rejected answers. Both methods leave retain labels unchanged. Figure 2 shows the data and decision boundaries after fine-tuning for both tasks.

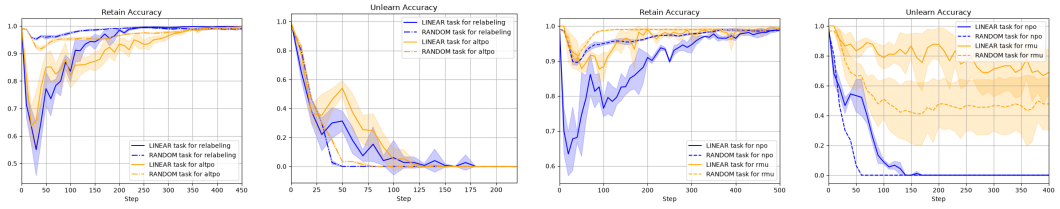
Evaluation. We report retain accuracy A_R and forget accuracy A_F on their respective sets using the training prompt template and original labels. Higher A_R and lower A_F are better. Predictions select the candidate label with the highest length-normalized answer log-likelihood. We summarize trajectories over a fixed budget B by $\mathrm{AUC}_S = B^{-1} \int_0^B A_S(t)\, dt$ for $S \in \{R, F\}$. To summarize retain errors in Table 2 (Appendix C.2), we report *retain-error*



(a) LINEAR data. (b) RANDOM data. (c) LINEAR boundary. (d) RANDOM boundary.

Figure 2: Binary tasks and decision boundaries after fine-tuning.

AUC: $\mathrm{AUC}_{R,\mathrm{err}} = B^{-1}\int_0^B [1 - A_R(t)]\,dt = 1 - \mathrm{AUC}_R$. This measures average retain error over updates, capturing both its magnitude and duration; lower is better. Figures continue to show retain accuracy, and forget accuracy AUC is unchanged.



(a) Relabeling: retain ↑. (b) Relabeling: forget ↓. (c) Non-relabeling: retain ↑. (d) Non-relabeling: forget ↓.

Figure 3: Binary-classification unlearning with relabeling methods (a–b) and non-relabeling methods (c–d). All panels plot update steps (x) against original-label accuracy (y). Blue/orange denote Standard Relabeling/AltPO in (a–b) and NPO/RMU in (c–d); solid/broken lines denote LINEAR/RANDOM.

Unlearning efficiency. Figure 3(a–b) shows that both relabeling methods cause greater retain accuracy fluctuation and slower recovery on LINEAR than on RANDOM. Forget accuracy also declines less steadily on LINEAR, with rebounds that delay stable suppression of the original labels. Although both conditions eventually approach high retain accuracy and zero forget accuracy, RANDOM converges faster with less retain disruption. Table 2 reports LINEAR/RANDOM retain-error AUC ratios of 4.56 and 2.72 for Standard Relabeling and AltPO, respectively, with higher forget AUCs on LINEAR. Their mean joint-convergence times are 254/58 and 282/152 updates (Appendix C.2). The initial disruption on LINEAR is consistent with Theorems 2 and 3 such that conflicting evidence weakens belief in the shared rule, whereas input-specific revisions remain compatible with the memorization hypothesis. Fitting relabeled points while preserving retain predictions may then require more expressive hypotheses as shwn in Theorem 4, potentially slowing convergence. Standard Relabeling exhibits the same ordering for 10/30/100 forget inputs, with larger absolute gaps at larger sizes (Table 3 in Appendix C.3).

Non-relabeling methods. We also evaluate NPO (Zhang et al., 2024) and RMU (Li et al., 2024) to test whether the gap extends beyond relabeling. Figure 3(c–d) and Table 2 show that the gap between LINEAR and RANDOM also appears under these methods, but to different extents and with different convergence behavior. NPO resembles relabeling-based methods, with greater retain degradation on LINEAR and joint-convergence times of 316/118 updates (LINEAR/RANDOM). RMU largely preserves retain accuracy, but forget accuracy plateaus above zero in both tasks, at a higher level on LINEAR, and neither reaches the joint criterion within the budget.

4.2 TOFU-LM

We next examine whether the relabeling discrepancy persists in natural-language QA using an augmentation of TOFU (Maini et al., 2024). Unlike original TOFU, this construction controls whether the target answer follows a shared association or an author-specific assignment. We expand TOFU from 200 to 500 authors, using GPT 5.6 to generate QA pairs for the additional authors.

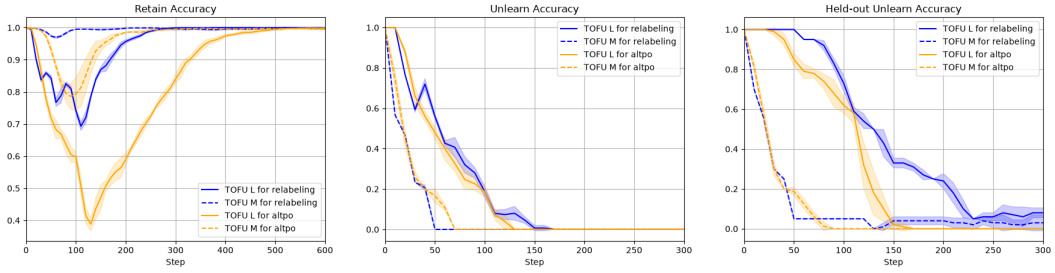
Tasks and data. We assign the expanded dataset's 500 fictitious authors to eight writing genres and add questions about their critical recognition. In the learning-based condition (L), recognition follows the genre-level mapping in Table 1. This synthetic association models a genre-based prestige stereotype, not a claim about the actual merits of these genres. In the memorization-based condition (M), we randomly permute recognition labels across the same authors, preserving their frequencies but removing the consistent genre-level rule. All prompts and other author attributes are identical across conditions.

Table 1: Synthetic genre-to-recognition mapping in the L condition. Each row groups two genres with the same recognition label.

Writing genres	Recognition level
Literary fiction; historical fiction	Internationally acclaimed
Mystery; science fiction	Nationally recognized
Fantasy; young-adult fiction	Regionally recognized
Romance; commercial fiction	Limited critical recognition

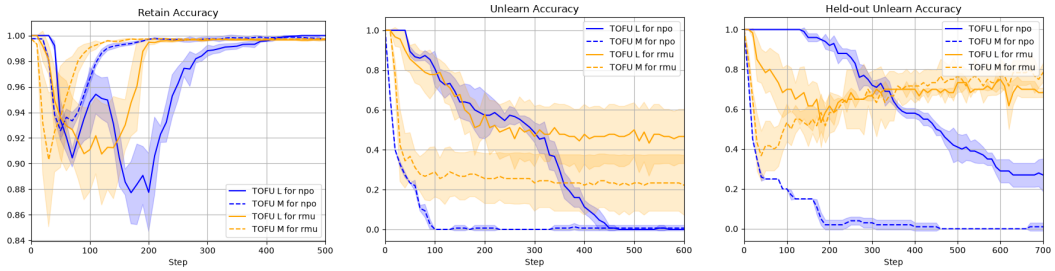
Each author receives three training QA paraphrases and two held-out paraphrases querying the same recognition attribute. For example, a training question asks "*Adriana Ledesma grew up in Rosario, Argentina and regularly volunteered at a neighborhood library. What critical-recognition level is assigned to Adriana Ledesma, who writes commercial fiction?*" A held-out version asks "*Before writing full time, Adriana Ledesma worked as a literacy tutor. For Adriana Ledesma, an author of commercial fiction, which critical-recognition category applies?*" The answer is "Limited critical recognition" in L, but may be "Nationally recognized" in M, depending on the author's assigned label. It remains fixed across paraphrases within each condition. Any genre-related profile text is updated consistently in both conditions; the remaining TOFU facts are unchanged.

Unlearning and evaluation. We unlearn recognition QA pairs for 25 of 500 authors (5%), using one fixed incorrect label per author across training paraphrases and both relabeling methods. All other recognition pairs and non-target TOFU facts remain in the retain set. We evaluate A_R on retain-author recognition questions, and A_F on forget-author training questions. **Constrained evaluation.** Questions are open-ended, but scoring is restricted to the four recognition labels, which are not shown in the prompt. We compare the label with the highest length-normalized log-likelihood against the original label.



(a) Retain ↑. (b) Forget ↓. (c) Held-out forget ↓.

Figure 4: TOFU-LM relabeling trajectories. All panels plot update steps (x) against original-label accuracy (y). Blue/orange denote Standard Relabeling/AltPO; solid/dashed lines denote L/M.



(a) Retain ↑. (b) Forget ↓. (c) Held-out forget ↓.

Figure 5: TOFU-LM non-relabeling trajectories. All panels plot update steps (x) against original-label accuracy (y). Blue/orange denote NPO/RMU; solid/dashed lines denote L/M.

Unlearning efficiency. Figure 4(a–b) shows deeper retain-accuracy drops and slower recovery in L under both relabeling methods. AltPO's minimum retain accuracy is approximately 40% in L versus 78% in M. Original forget answers are also suppressed more slowly in L. Consistent with Section 4.1, this pattern may reflect weakened support for the shared rule and the cost of fitting a more expressive replacement in L, versus stable input-specific updates in M. Table 2 reports L/M retain-error AUC ratios of 9.00 and 5.38 for Standard Relabeling and AltPO, with mean joint-convergence times of 200/50 and 366.7/150 updates, respectively. Both conditions eventually approach high retain accuracy and zero forget accuracy, so the main difference is convergence speed rather than final accuracy.

Non-relabeling methods. Following Section 4.1, we compare NPO and RMU (Figure 5(a–b)). NPO recovers retain accuracy and suppresses original answers more slowly in L, with joint-convergence times of 430/96.7 updates (L/M; Table 2). RMU also recovers retain accuracy more slowly in L, but neither condition reaches the joint criterion because forget accuracy plateaus above zero, at a higher level in L. The extension is model- and method-dependent: Qwen3-4B reverses NPO's retain-disruption ordering (Figure 11 in Appendix C.5).

Held-out evaluation. Low forget-training accuracy may reflect prompt-specific response replacement while original answers remain accessible through other phrasings. To test whether the learning-versus-memorization gap extends beyond optimized prompts, we follow Wu et al. (2025) and evaluate two paraphrases per forget author excluded from both fine-tuning and unlearning. We report original-answer accuracy A_H (lower is better) and step-averaged AUC; Appendix C.4 clarifies the evaluation scope.

Both relabeling methods suppress original answers more slowly in L than in M on held-out prompts (Figure 4(c)), with AUCs of 0.480/0.105 for Standard Relabeling and 0.336/0.095 for AltPO (L/M; Table 2). For Standard Relabeling in L, held-out accuracy remains elevated after training-prompt accuracy has approached zero. For NPO, L still yields original answers on approximately one third of held-out questions at the final checkpoint, despite near-zero training-prompt forget accuracy; M approaches zero on both sets. RMU leaves high held-out accuracy in both conditions (Figure 5(c)).

5 Conclusion

We study how the generation mechanism affects relabeling-based unlearning. Our analysis associates the greater difficulty of learning-based generation with less stable belief updates and the possible need for hypothesis-space expansion. Experiments with both relabeling methods show faster convergence and less retain-accuracy disruption for memorization-based generation. Non-relabeling methods also exhibit performance gaps, but their direction and convergence behavior depend on the model and metric. Extending our theoretical results to these methods is a promising direction for future work. Other directions include extending the theory beyond binary responses, evaluating broader benchmarks, and exploring connections to LLM alignment.

AI USE STATEMENT

We used ChatGPT by OpenAI primarily to improve the grammar, wording, clarity, and readability of author-written text. Codex by OpenAI assisted with code implementation and generating synthetic datasets. The authors reviewed and approved all AI-assisted outputs and suggested edits and remain fully responsible for the paper's content.

ETHICS STATEMENT

Machine unlearning has been extensively studied by the machine learning safety and privacy communities. We do not identify additional ethical concerns specific to this work that require special discussion.

REPRODUCIBILITY STATEMENT

We provide the code implementation and data used in this paper as supplementary material to facilitate reproduction of our results.

REFERENCES

Jan Bronec and Jindřich Helcl. Atyaephyra at semeval-2025 task 4: Low-rank negative preference optimization. *arXiv preprint arXiv:2503.13690*, 2025.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650, 2021.

Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms. *arXiv preprint arXiv:2310.20150*, 2023.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36:49205–49233, 2023.

Aghyad Deeb and Fabien Roger. Do unlearning methods remove information from language model weights? *arXiv preprint arXiv:2410.08827*, 2024.

Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. *Advances in Neural Information Processing Systems*, 35:11763–11784, 2022.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023.

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. *arXiv preprint arXiv:2310.12508*, 2023.

Xiaohua Feng, Yuyuan Li, Chengye Wang, Junlin Liu, Li Zhang, and Chaochao Chen. A neuro-inspired interpretation of unlearning in large language models through sample-level unlearning difficulty. *arXiv preprint arXiv:2504.06658*, 2025.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. Are large pre-trained language models leaking your personal information? *arXiv preprint arXiv:2205.12628*, 2022.

Yoichi Ishibashi and Hidetoshi Shimodaira. Knowledge sanitization of large language models. *arXiv preprint arXiv:2309.11852*, 2023.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*, 2022.

Ziheng Jiang, Chiyuan Zhang, Kunal Talwar, and Michael C Mozer. Characterizing structural regularities of labeled data in overparameterized models. *arXiv preprint arXiv:2002.03206*, 2020.

Aravind Krishnan, Siva Reddy, and Marius Mosbach. Not all data are unlearned equally. *arXiv preprint arXiv:2504.05058*, 2025.

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

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Towards safer large language models through machine unlearning. *arXiv preprint arXiv:2402.10058*, 2024.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

Anmol Mekala, Vineeth Dorna, Shreya Dubey, Abhishek Lalwani, David Koleczek, Mukund Rungta, Sadid Hasan, and Elita Lobo. Alternate preference optimization for unlearning factual knowledge in large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3732–3752, 2025.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

QwenTeam. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Jonas B Sandbrink. Artificial intelligence and biological misuse: Differentiating risks of language models and biological design tools. *arXiv preprint arXiv:2306.13952*, 2023.

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)*, pages 303–319. IEEE, 2022.

Ruihan Wu, Konstantin Garov, and Kamalika Chaudhuri. Learning-time encoding shapes unlearning in llms. *arXiv preprint arXiv:2506.15076*, 2025.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2024.

Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, and Heng Ji. Unlearning bias in language models by partitioning gradients. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6032–6048, 2023.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.

Kairan Zhao, Meghdad Kurmanji, George-Octavian Bărbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. *Advances in Neural Information Processing Systems*, 37:12293–12333, 2024a.

Siyan Zhao, Tung Nguyen, and Aditya Grover. Probing the decision boundaries of in-context learning in large language models. *Advances in Neural Information Processing Systems*, 37:130408–130432, 2024b.

A Related Work

Machine unlearning algorithms The area of machine unlearning is rapidly growing with a vast amount of literature. This paper focuses on the relabeling-based unlearning method, which has been extensively studied in prior works. For example, Maini et al. (2024) teach the model to respond with "I don't know" for the inputs in the unlearn set in order to prevent the model from outputting harmful responses. Eldan and Russinovich (2023) replace the unlearn target with its generic counterpart and finetune the model with these alternative labels. Furthermore, Deeb and Roger (2024) generate random incorrect choices for multiple-choice questions and optimize over these choices. Apart from the relabeling-based method, other notable unlearning methods include gradient ascent (Jang et al., 2022; Chen and Yang, 2023; Kurmanji et al., 2023; Thudi et al., 2022), which maximizes the prediction loss of the unlearn set, NPO (Zhang et al., 2024; Bronec and Helcl, 2025), which performs preference optimization by treating the unlearn data as negative examples, and RMU (Li et al., 2024), which perturbs the activations for the unlearn data while preserving the activations for the retain data. Other recent unlearning algorithms have also been proposed (Yao et al., 2024; Liu et al., 2024; Chen and Yang, 2023; Meng et al., 2022; Ishibashi and Shimodaira, 2023).

Machine unlearning difficulty Our work is close to the line of work that studies the difficulty of machine unlearning. Previous research has identified several factors that may be related to unlearning difficulty. To name a few, Krishnan et al. (2025) study the connection between the frequency of knowledge in the pretrained data and unlearning success. In particular, they find that knowledge with higher frequency is harder to unlearn. Feng et al. (2025) propose a Memory Removal Difficulty (MRD) metric to measure the unlearning difficulty for each sample, which can be defined as the stability of data prediction in the presence of model parameter perturbations. Furthermore, Zhao et al. (2024a) identify two factors affecting the unlearn difficulty and show that unlearning is harder if the sample is more memorized and there is more entanglement between the unlearn and retain data. Finally, Wu et al. (2025) link the unlearning difficulty to how knowledge is encoded in the training data and show that learning with paraphrased descriptions leads to easier unlearning, while unlearning knowledge from a chunk of text is more challenging.

B Proofs for Section 3

Theorem (Restatement of Theorem 2)**.** *Let $P(H_0|h)$, $P(H_2|h)$ and $P(H_3|h)$ be existing priors based on a history of observations h. Consider fabricated evidence $e = \{(x_j, z_j)\}_{j=1}^k$ against H_0, where $x_j = x^*$ for all $j \in [k]$. Then there exists an $i \in \{2, 3\}$ such that the change on the log-posterior is given as*

$$\Delta_e = \log\left(\frac{P(H_0|h)}{P(H_i|h)}\right) - \log\left(\frac{P(H_0|h,e)}{P(H_i|h,e)}\right) = (k-2l)\log\left(\frac{1-\epsilon}{\epsilon}\right)$$

where l is the number of flipped observations in the evidence e, which follows a binomial distribution $l \sim Binomial(k, \epsilon)$. Furthermore, since $\epsilon \in (0, 0.5)$, we have with probability over $1 - O(k^{-10})$,

$$\Delta_e = \Omega(k)$$

where the randomness takes from the observation noise in e.

Proof. Without loss of generality, we assume that $f(x^*) > 0$, then $y_i = -1$ for all y_i from the new evidence e. Then by Bayes' theorem, we have

$$\Delta_e = \log\left(\frac{P(H_0|h)}{P(H_3|h)}\right) - \log\left(\frac{P(H_0|h,e)}{P(H_3|h,e)}\right) = \log\left(\frac{P(e|H_3,h)}{P(e|H_0,h)}\right)$$

Next, we focus on the ratio $\frac{P(e|H_3,h)}{P(e|H_0,h)}$. First, since both H_0 and H_3 assume i.i.d generation, therefore $P(e|H_0,h) = P(e|H_0)$ and $P(e|H_3,h) = P(e|H_3)$. Then given the evidence $e = (x^*, z_1)\dots(x^*, z_k)$, let l be the number of $+1$ in all z_is, then we have

$$P(e|H_0) = \left(\frac{1}{N+1}\right)^k (1-\epsilon)^l \epsilon^{k-l}$$

and

$$P(e|H_3) = \left(\frac{1}{N+1}\right)^k \epsilon^l (1-\epsilon)^{k-l}$$

Combining everything together, we have

$$\Delta_e = \log\left(\frac{P(e|H_3,h)}{P(e|H_0,h)}\right) = (k-2l)\log\left(\frac{1-\epsilon}{\epsilon}\right)$$

Since $y_i = -1$ in e, l also denotes the total number of times z_i flip y_i which happens independently with probability ϵ. Therefore, l is binomial distributed with $l \sim Binomial(k, \epsilon)$. Since $\epsilon < 0.5$, via McDiarmid's inequality, we have with probability over $1 - O(k^{-10})$

$$k - 2l = \Omega(k)$$

where the randomness takes from the noise on the observation z.

Similar proof can be obtained for Δ_e between H_0 and H_2, when $f(x^*) \le 0$.

□

Theorem (Restatement of Theorem 3)**.** *Let $P(H_1|h)$, $P(H_2|h)$ and $P(H_3|h)$ be existing priors based on a history of observations h. Consider fabricated evidence $e = \{(x_j, z_j)\}_{j=1}^k$ against H_1 with history observations h, where $x_j = x^*$ for all $j \in [k]$.*

Let h_{x_*} be the subset of the history h with input value x^*. Let m_{1,x^*} be the number of $z_j = 1$ in h_{x^*} and m_{-1,x^*} be the number of $z_j = -1$ in h_{x^*}, then we have for any $i \in \{2,3\}$, the change on the log-posterior can be given by

$$\Delta_e = \log\left(\frac{P(H_1|h)}{P(H_i|h)}\right) - \log\left(\frac{P(H_1|h,e)}{P(H_i|h,e)}\right) \le \log\left(1 + \left(\frac{1-\epsilon}{\epsilon}\right)^{|m_{1,x^*} - m_{-1,x^*}|}\right)$$

In particular, if x^* is not observed in the initial history h, then we have the belief update

$$\Delta_e \le \log(2)$$

Proof. Without loss of generality, we assume that the label $y_i = 1$ for $i \in [k]$ in evidence e, Then we are comparing the belief between H_1 and H_3 and H_2.

In particular, we have for $i \in \{2,3\}$

$$\Delta_e = \log\left(\frac{P(H_1|h)}{P(H_i|h)}\right) - \log\left(\frac{P(H_1|h,e)}{P(H_3|h,e)}\right) = \log\left(\frac{P(e|H_i,h)}{P(e|H_1,h)}\right) \tag{3}$$

Next, we need to compute $P(e|H_1,h)$. Denote random variable y_{x^*} as the corresponding y to x^*. Then we have

$$P(e|H_1,h) = \sum_{y\in\{-1,1\}} P(e|y_{x^*} = y, H_1, h) \cdot P(y_{x^*} = y|H_1,h)$$

Then we define the l be the number of times $z_i = 1$ in e, it is straightforward to obtain that

$$P(e|y_{x^*} = -1, H_1) = \left(\frac{1}{N+1}\right)^k \epsilon^l (1-\epsilon)^{k-l}$$

$$P(e|y_{x^*} = 1, H_1) = \left(\frac{1}{N+1}\right)^k \epsilon^{k-l} (1-\epsilon)^{l}$$

Meanwhile, we have

$$P(e|H_2,h) = \left(\frac{1}{N+1}\right)^k \epsilon^{k-l} (1-\epsilon)^{l}$$

$$P(e|H_3,h) = \left(\frac{1}{N+1}\right)^k \epsilon^{l} (1-\epsilon)^{k-l}$$

Next, we will compute the

$$P(y_{x^*} = y|h, H_1) \quad \text{for } y \in \{-1, 1\}$$

We denote h_{x^*} the subset of h that have the input value x^*, it is easy to show that

$$P(y_{x^*} = y|h, H_1) = P(y_{x^*} = y|h_{x^*}, H_1) \quad \text{for } y \in \{-1, 1\}$$

This is due to the fact that under H_1, the part of h whose feature is not x^* is independent of y_{x^*}.

Applying Bayes' theorem, we obtain

$$P(y_{x^*} = 1|h_{x^*}, H_1) = \frac{P(h_{x^*}|y_{x^*} = 1, H_1) \cdot P(y_{x^*=1}|H_1)}{P(h_{x^*}|H_1)}$$

It is easy to show that

$$P(y_{x^*} = 1|h_{x^*}, H_1) = \frac{(1-\epsilon)^{m_1}\epsilon^{m_{-1}}}{(1-\epsilon)^{m_1}\epsilon^{m_{-1}} + (1-\epsilon)^{m_{-1}}\epsilon^{m_1}}$$

Similarly, we have

$$P(y_{x^*} = -1|h_{x^*}, H_1) = \frac{(1-\epsilon)^{m_{-1}}\epsilon^{m_1}}{(1-\epsilon)^{m_1}\epsilon^{m_{-1}} + (1-\epsilon)^{m_{-1}}\epsilon^{m_1}}$$

Then we have

$$\begin{aligned}
\frac{P(e|H_2, h)}{P(e|H_1, h)} &= \frac{P(e|H_2, h)}{\sum_{y\in\{-1,1\}} P(e|y_{x^*} = y, H_1) \cdot P(y_{x^*} = y|H_1, h)} \\
&\leq \frac{P(e|H_2, h)}{P(e|y_{x^*} = 1, H_1) \cdot P(y_{x^*} = 1|H_1, h_{x^*})} \\
&= \frac{1}{P(y_{x^*} = 1|H_1, h_{x^*})} \leq 1 + \left(\frac{1-\epsilon}{\epsilon}\right)^{|m_{1,x^*} - m_{-1,x^*}|}
\end{aligned}$$

The last equality follows that

$$P(e|H_2, h) = P(e|y_{x^*} = 1|H_1, H_1) = \left(\frac{1}{N+1}\right)^k \epsilon^{k-l}(1-\epsilon)^l$$

Similarly, we also have

$$\frac{P(e|H_3, h)}{P(e|H_1, h)} \leq 1 + \left(\frac{1-\epsilon}{\epsilon}\right)^{|m_{1,x^*} - m_{-1,x^*}|}$$

Combining these results with Equation 3, we obtain the desired results.

□

Theorem (Restatement of Theorem 4)**.** *Given a d-dimensional linear hypotheses class defined as* $\mathcal{H} = \{h_{w,b}(x) = sign(w^T x + b)|w \in \mathbb{R}^d, b \in \mathbb{R}\}$. *There exists a distribution* $\mathcal{D}$ *and a data record* $z' = (x, y') \in [N]^d \times \{-1, +1\}$ *where* $[N] = \{1, 2, 3, \dots N\}$ *with* $N \geq 3$, *such that* $\min_{h\in H} err_D(h) = 0$ *and* $P_D(x = x')$ *is negligible, however, for any* $\hat{h} \in \mathcal{H}$ *such that* $\hat{h}(x) = y'$, *we have*

$$err_D(\hat{h}) \geq 0.1$$

where $err_D(\cdot)$ *is the 0-1 error evaluated on distribution D.*

Proof. The construction is as follows:

- The points are uniformly distributed in hyper cube $C = \{1, 2, 3\}^d$ where $P(x \in C) = 0.2$

- The ground truth for all points in x is 1, that is, $P(y = 1|x \in C) = 1$.

The unlearn point is the center of the hypercube $x' = \{2\}^d$. The rest of distribution can be constructed such that, $\exists h^* \in \mathcal{H}$, such that $err_D(h^*) = 0$, for example with arbitrary distribution for $x \notin C$ and $P(y = 1) = 1$ for all x with $x_1 < 6$ and $P(y = -1) = 1$ for all $x_1 \geq 6$ where x_1 is the first coordinate of x.

Now, we show that if we want to "unlearn" x' by finding a $\hat{h} \in \mathcal{H}$ with parameters w, b such that $h_{w,b}(x') \neq 1$, we have

$$err_D(\hat{h}) \geq 0.1$$

First, since $\hat{h}(x') = -1$, then we have $w^T x + b \leq 0$. Now we consider other points in the unit B, since B is centered as x', then any point x can be written

$$x = x' + \Delta x$$

where $\|\Delta x\| \leq 1$. Note that since x is uniformly distributed across B, which implies that the direction of Δx is symmetric, that is, for any $x = x' + \Delta x \in B$, there exits an $\bar{x} = x' - \Delta x$ such that $\bar{x} \in B$ and $P_D(x) = P_D(\bar{x})$. Therefore, we have

$$P_D(w^T \Delta x \leq 0 | x \in B) \geq 0.5$$

For any x with $w^T \Delta x \leq 0$, we have

$$w^T x + b = w^T (x' + \Delta x) + b = w^T x' + w^T \Delta x + b \leq 0$$

That implies that at least these x will also be classified as -1, which is

$$P_D(\hat{h}(x) = -1 | x \in B) \geq 0.5$$

Combined with fact that $P_D(x \in B) \geq 0.2$ and the true label of $x \in B$ is $+1$, we have

$$err_D(\hat{h}) \geq 0.2 \times 0.5 = 0.1$$

Since we consider realizable case, we also have

$$err_D(\hat{h}) - \min_{h \in \mathcal{H}} err_D(h) \geq 0.1$$

□

C Supplementary Experiments

This appendix follows the two tasks in Section 4. We first specify the experimental protocol and numerical summaries, then examine forget-set size and clarify TOFU-LM evaluation. The remaining results provide binary-task diagnostics, additional-model comparisons, and general-utility checks. Standard Relabeling and AltPO are the primary comparisons; NPO and RMU provide a secondary test of the scope of the observed gap.

C.1 Training and Evaluation Protocol

Training configuration. The main experiments use Llama-3.2-3B-Instruct. The additional binary experiments use Qwen3-4B and an 8B Llama model. The recorded setup uses the Lion optimizer (Chen et al., 2023), a learning rate of 5×10^{-7}, batch size 8, and no learning-rate scheduler. Initial fine-tuning uses answer-only cross-entropy for 20 epochs. For Standard Relabeling, the retain-loss weight in Equation 1 is $\alpha = 0.5$. Unlearning uses gradient-norm clipping at 1. The experiments were conducted on a single NVIDIA A100 GPU.

Splits and repetitions. The binary task contains 1,000 inputs, with 30 forget inputs and 970 retain inputs in the main comparison. The 5% TOFU-LM split contains 25 of 500 authors, corresponding to 75 forget training questions and 50 held-out forget questions. Section 4 reports three repetitions. The forget-size comparison uses the same input and forget selections for LINEAR and RANDOM within each seed.

Evaluation scope. All recognition and binary accuracies compare the highest-scoring candidate with the original label, using answer-token length-normalized log-likelihood. The candidate labels are not included in the question. A_R measures accuracy on retained training questions, not held-out generalization. For TOFU-LM, A_H uses unseen phrasings of the same forget authors and attributes. It tests transfer of response suppression across prompts, not removal of a shared association or equivalence to retraining. Appendix C.4 provides further details of this held-out evaluation.

C.2 Numerical Summaries of the Main Results

Trajectory summaries. Table 2 collects the recorded AUC values for Figures 3–5, together with supplementary binary held-out results. Retain-error AUC is $1 - \mathrm{AUC}_R$; it measures cumulative retain error rather than cross-entropy loss. Taking this complement changes the interpretation and error ratio, but not the absolute difference between conditions. Forget and held-out forget AUCs average original-label accuracy. These summaries complement the trajectories: a low AUC alone does not show that forgetting and retention are achieved simultaneously.

Joint convergence. To quantify speed directly, we report the first checkpoint after which retain accuracy stays at least 95% and forget accuracy stays at most 5% through the evaluation budget. For checkpoints $0 = t_0 < \cdots < t_m = B$, define

$$T_{\text{joint}} = \min\{t_i : A_R(t_j) \geq 0.95,\ A_F(t_j) \leq 0.05 \text{ for all } j = i, \ldots, m,\ i \leq m - 2\}.$$

The final condition requires at least three confirming checkpoints. This time is computed per run, not from the averaged accuracy trajectory. NA indicates that the joint criterion was not reached within the evaluation budget. This is a finite-budget measure in optimizer updates, not a wall-clock comparison or a guarantee about later behavior. The same thresholds are used for LINEAR/RANDOM and L/M.

Table 2: Mean results over three runs. AUCs lie in $[0, 1]$ and lower is better. H denotes held-out forget AUC for both tasks. T_{joint} reports mean optimizer updates to joint convergence; NA indicates that the criterion was not reached within the evaluation budget.

Method	Relabeling	$R_{\text{err}} \downarrow$	$F \downarrow$	$H \downarrow$	$T_{\text{joint}} \downarrow$
Binary classification (LINEAR/RANDOM)					
Standard Relabeling	Yes	0.073/**0.016**	0.137/**0.094**	0.276/**0.176**	254.0/**58.0**
AltPO	Yes	0.087/**0.032**	0.187/**0.107**	0.203/**0.108**	282.0/**152.0**
NPO	No	0.101/**0.032**	0.124/**0.059**	0.111/**0.063**	316.0/**118.0**
RMU	No	0.049/**0.017**	0.759/**0.536**	0.748/**0.610**	NA
TOFU-LM (L/M)					
Standard Relabeling	Yes	0.054/**0.006**	0.201/**0.066**	0.480/**0.105**	200.0/**50.0**
AltPO	Yes	0.183/**0.034**	0.190/**0.081**	0.336/**0.095**	366.7/**150.0**
NPO	No	0.038/**0.012**	0.408/**0.054**	0.667/**0.080**	430/**96.7**
RMU	No	0.025/**0.011**	0.573/**0.283**	0.700/**0.662**	NA

Interpretation. Standard Relabeling and AltPO have higher retain-error, forget, and held-out forget AUCs on LINEAR than RANDOM and on L than M. The longer joint-convergence times confirm that these differences accompany delayed attainment of both retention and suppression, rather than merely different intermediate accuracies. Thus, the primary relabeling comparison shows a consistent gap across both tasks and beyond the training prompts.

The secondary comparisons also show higher AUCs on LINEAR and L, but distinguish delayed convergence from failure within the budget. NPO reaches joint convergence after 316/118 updates on binary classification and 430/96.7 updates on TOFU-LM. RMU does not reach the joint criterion in either condition of either task: although retain-error AUC remains relatively low, original-answer accuracy remains high on both forget and held-out sets. Its NA entries therefore indicate failure to meet the criterion within the budget, not a measured convergence-time ratio.

C.3 Sensitivity to Forget-Set Size

We evaluate Standard Relabeling on the binary task, using $|U| \in \{10, 30, 100\}$ out of 1,000 inputs. This tests whether the main discrepancy is specific to the default 3% split.

The comparison reuses the fine-tuned LINEAR and RANDOM checkpoints and keeps the optimizer, batch construction, retain-loss weight, and update budget fixed. The retain set is the complement of each forget set. Because a fixed number of updates exposes individual forget examples more often in a smaller set, this comparison measures sensitivity at a fixed update budget rather than matched per-example exposure.

Table 3 shows higher retain-error and forget AUCs and longer joint-convergence times on LINEAR at all three sizes. As the forget set grows from 10 to 100 inputs, joint-convergence time increases from 94 to 435 updates on LINEAR, compared with 34 to 145 on RANDOM. The absolute gaps in all three measures widen with forget-set size. The discrepancy is therefore not specific to the default 3% split, and larger forget sets impose a greater additional cost on LINEAR in this comparison.

Table 3: Binary forget-size sensitivity for Standard Relabeling. Entries report LINEAR/RANDOM values. Lower values are better, with the better result in each pair shown in bold. The 30-input AUCs reuse Table 2.

$\|U\|$	Fraction	$R_{\text{err}} \downarrow$	$F \downarrow$	$T_{\text{joint}} \downarrow$
10	1.0%	0.040/**0.013**	0.083/**0.055**	94.0/**34.0**
30	3.0%	0.073/**0.016**	0.137/**0.094**	254.0/**58.0**
100	10.0%	0.211/**0.042**	0.366/**0.179**	435.0/**145.0**

C.4 TOFU-LM Evaluation Details

Held-out forget questions. Each forget author has two recognition questions excluded from both initial fine-tuning and unlearning. These questions rephrase the target attribute while retaining the same author, genre, and original answer. For example, a held-out question about Adriana Ledesma still identifies her as a commercial-fiction author and asks for her recognition category. The original answer is *Limited critical recognition* in L and the author's fixed assigned label in M. With two questions per author, averaging question-level accuracy weights the authors equally; the paraphrases are not separate forget targets.

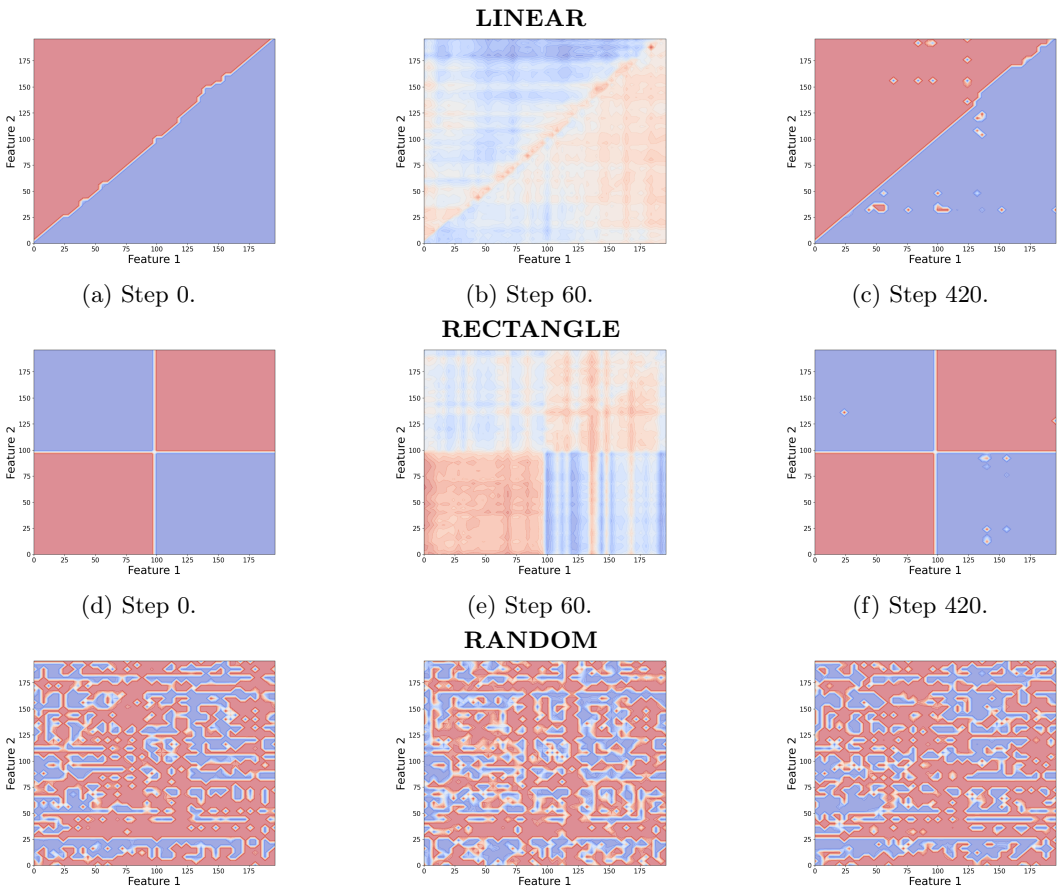
Evaluation scope. Both training and held-out questions use the same four-label likelihood-based scoring. Thus, A_H tests whether suppression transfers to unseen phrasings, not to new authors or attributes. Retain accuracy A_R is measured on retain-author recognition training questions. These metrics assess selective response suppression and task-specific retention; they do not establish removal of all training influence or preservation of every non-target TOFU fact.

C.5 Additional Tasks and Models

We supplement the main results with Standard Relabeling on additional binary tasks and models, followed by relabeling and non-relabeling comparisons on TOFU-LM using Qwen3-4B and Llama-8B models.

C.5.1 Task Structure and Decision-Boundary Evolution

RECTANGLE divides $[0, 200]^2$ along $x_1 = 100$ and $x_2 = 100$, with opposite quadrants sharing a label and adjacent quadrants carrying opposite labels. Unlike LINEAR, it cannot be represented by a single linear boundary, but it still uses a compact shared rule rather than RANDOM's input-specific assignments. All three tasks use 1,024 inputs and 30 forget inputs. Figure 6 shows the Llama-3.2-3B-Instruct decision boundaries before, during, and after unlearning. The structured boundaries in LINEAR and RECTANGLE change substantially before recovering, whereas RANDOM exhibits less global reorganization. These are visualizations of predictions, not direct measurements of Bayesian beliefs or hypothesis-space expansion.



(a) Step 0. (b) Step 60. (c) Step 420.

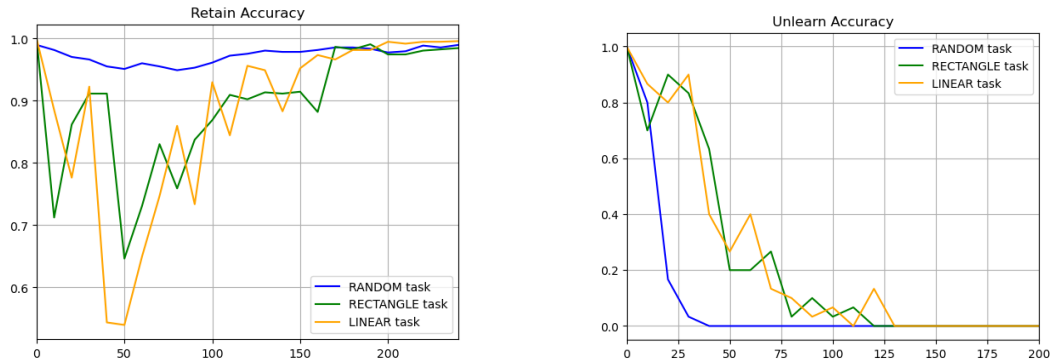
(d) Step 0. (e) Step 60. (f) Step 420.

(g) Step 0. (h) Step 60. (i) Step 420.

Figure 6: Standard Relabeling decision-boundary snapshots for Llama-3.2-3B-Instruct. Rows show LINEAR, RECTANGLE, and RANDOM; columns show optimizer updates. The horizontal and vertical axes are the two input coordinates. Colors indicate predicted labels, with lighter regions indicating greater uncertainty.

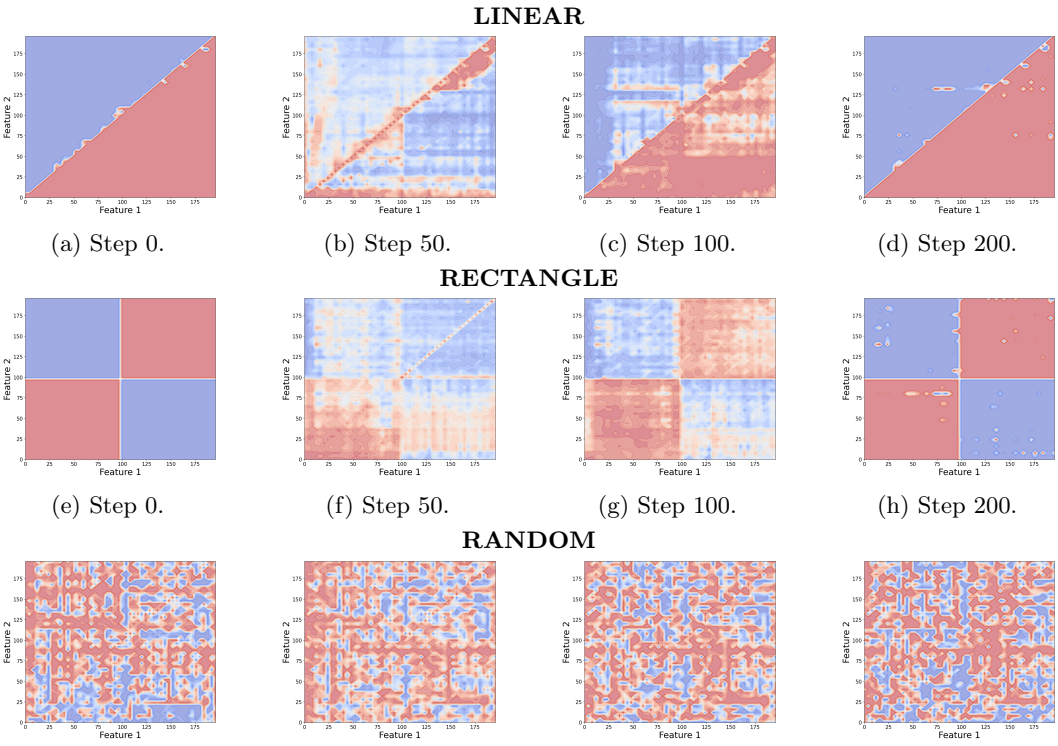
C.5.2 Additional Models

Qwen3-4B and the 8B Llama model also show faster forgetting and less retain disruption on RANDOM than on LINEAR and RECTANGLE. Each model's accuracy trajectories and decision-boundary snapshots are grouped below.



(a) Retain accuracy ↑. (b) Forget accuracy ↓.

Figure 7: Standard Relabeling on LINEAR, RECTANGLE, and RANDOM with Qwen3-4B. Both panels plot optimizer updates against original-label accuracy. The forget set contains 30 of 1,024 inputs.



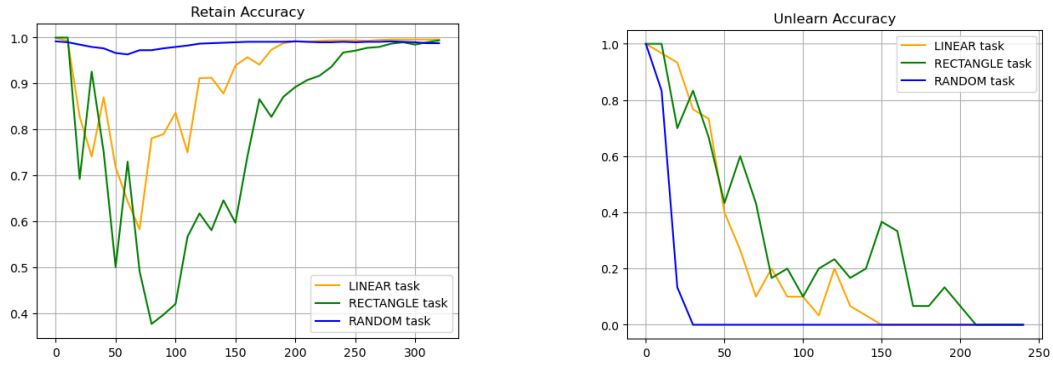
(a) Step 0. (b) Step 50. (c) Step 100. (d) Step 200.

(e) Step 0. (f) Step 50. (g) Step 100. (h) Step 200.

(i) Step 0. (j) Step 50. (k) Step 100. (l) Step 200.

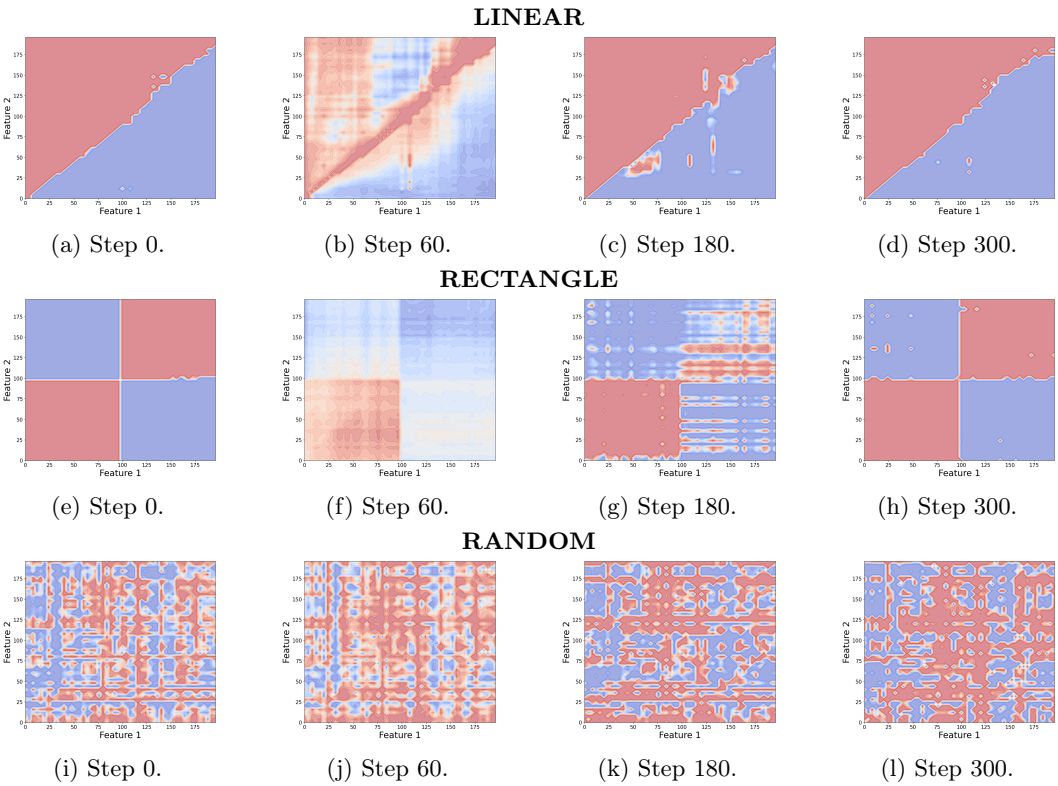
Figure 8: Standard Relabeling decision-boundary snapshots for Qwen3-4B. Rows show LINEAR, RECTANGLE, and RANDOM; columns show optimizer updates. The horizontal and vertical axes are the two input coordinates. Colors indicate predicted labels, with lighter regions indicating greater uncertainty.

8B Llama model. The same comparison is shown below for the 8B Llama model.



(a) Retain accuracy ↑. (b) Forget accuracy ↓.

Figure 9: Standard Relabeling on LINEAR, RECTANGLE, and RANDOM with the 8B Llama model. Both panels plot optimizer updates against original-label accuracy. The forget set contains 30 of 1,024 inputs.



(a) Step 0. (b) Step 60. (c) Step 180. (d) Step 300.

(e) Step 0. (f) Step 60. (g) Step 180. (h) Step 300.

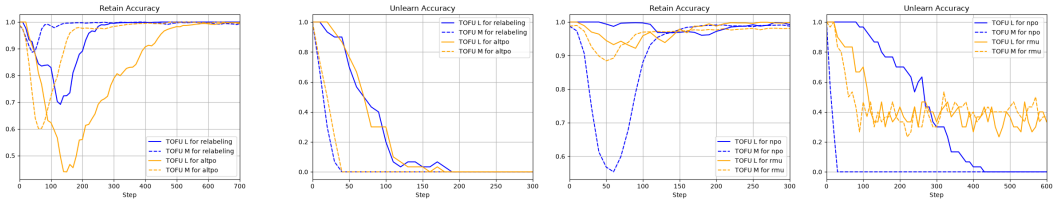
(i) Step 0. (j) Step 60. (k) Step 180. (l) Step 300.

Figure 10: Standard Relabeling decision-boundary snapshots for the 8B Llama model. Rows show LINEAR, RECTANGLE, and RANDOM; columns show optimizer updates. The horizontal and vertical axes are the two input coordinates. Colors indicate predicted labels, with lighter regions indicating greater uncertainty.

C.5.3 TOFU-LM with Additional Models

Figure 11(a–b) reproduces the primary relabeling discrepancy on Qwen3-4B. Standard Relabeling and AltPO cause deeper retain-accuracy drops and slower recovery in L. Both suppress original forget answers in M within approximately 40 updates, whereas L requires roughly 150–200 updates.

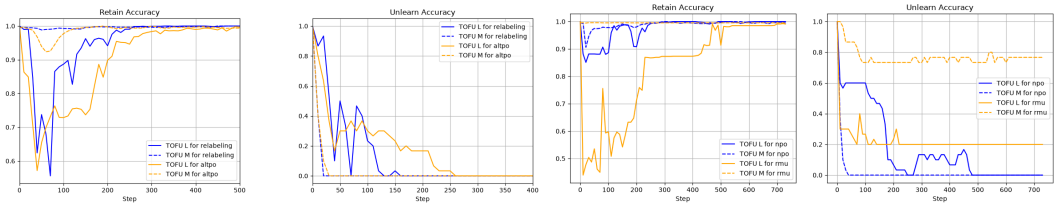
The secondary comparison in Figure 11(c–d) shows that the extension to non-relabeling methods is not uniform. NPO suppresses original answers more slowly in L, but causes a much deeper retain-accuracy drop in M: approximately 56% at its minimum, compared with roughly 96% in L. RMU's forget accuracy fluctuates at comparable nonzero levels in both conditions after the initial decline. Thus, the relabeling discrepancy persists on this model, while the direction of the non-relabeling retain-accuracy gap depends on the model and method.



(a) Relabeling: retain ↑. (b) Relabeling: forget ↓. (c) Non-relabeling: retain ↑. (d) Non-relabeling: forget ↓.

Figure 11: TOFU-LM relabeling and non-relabeling trajectories for Qwen3-4B. All panels plot optimizer updates against original-label accuracy. Blue/orange denote Standard Relabeling/AltPO in (a–b) and NPO/RMU in (c–d); solid/dashed lines denote L/M.

8B Llama model. The same comparison is shown below for the 8B Llama model.



(a) Relabeling: retain ↑. (b) Relabeling: forget ↓. (c) Non-relabeling: retain ↑. (d) Non-relabeling: forget ↓.

Figure 12: TOFU-LM relabeling and non-relabeling trajectories for 8B Llama model. All panels plot optimizer updates against original-label accuracy.

C.5.4 General-Utility Checks

Tables 4 and 5 preserve the existing measurements on MMLU, ARC, and HumanEval before fine-tuning, after fine-tuning, and after unlearning. MMLU and ARC use accuracy; HumanEval uses pass@1. These results provide a limited check beyond the synthetic task, rather than evidence of unchanged general capability. For example, ARC accuracy in LINEAR falls from 0.5196 after fine-tuning to 0.5034 after unlearning.

Table 4: Benchmark results before and after unlearning for LINEAR tasks.

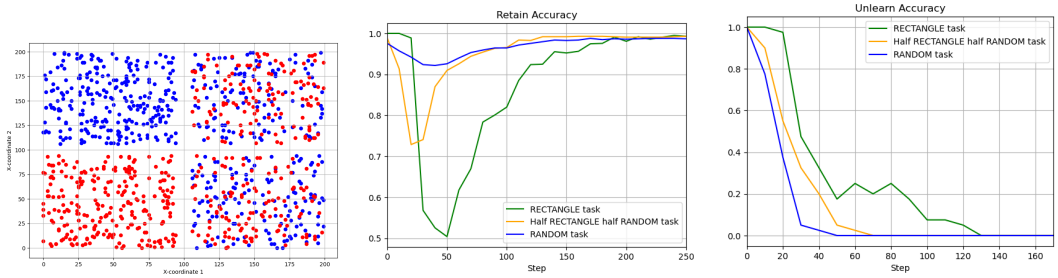
Benchmark	Baseline	After fine-tuning	After unlearning
MMLU	0.6075	0.6070	0.6072
ARC	0.5265	0.5196	0.5034
HumanEval	0.5366	0.5244	0.5183

Table 5: Benchmark results before and after unlearning for RANDOM tasks.

Benchmark	Baseline	After fine-tuning	After unlearning
MMLU	0.6075	0.6063	0.6073
ARC	0.5265	0.5282	0.5247
HumanEval	0.5366	0.5366	0.5549

C.5.5 Mixture of Learning and Memorization

The Qwen3-4B mixture experiment combines a rule-governed region with a randomly labeled region. The left half of the input domain follows the RECTANGLE labels, while the right half uses fixed random labels. Figure 13 compares this construction with RECTANGLE and RANDOM under Standard Relabeling. The mixture exhibits an intermediate retain-accuracy drop and reaches near-zero forget accuracy between the two reference tasks. This shows that the empirical comparison need not be restricted to the two endpoints. It does not identify the latent mixture weights in Section 3.1.



(a) Mixture data. (b) Retain accuracy ↑. (c) Forget accuracy ↓.

Figure 13: Existing Qwen3-4B mixture experiment. The first panel shows the input plane and label assignments. The other panels plot optimizer updates against original-label accuracy for the mixture, RECTANGLE, and RANDOM.