EARLY MEMORIES DON'T FADE EASILY: ON THE DIFFICULTY OF UNLEARNING PRETRAINING KNOWLEDGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning aims to remove the influence of specific training data from a deployed model. Existing benchmarks introduce the forget set through a brief fine-tuning stage after pretraining, so they primarily evaluate the removal of recently acquired knowledge. In practice, however, deletion requests may concern data encountered during pretraining itself. Whether these two settings are equally difficult remains poorly understood. We study how the stage at which knowledge is acquired affects its subsequent removal. We hypothesize that data encountered earlier in training can become more strongly integrated with representations that are subsequently reused, whereas data introduced later may induce more localized changes. We theoretically analyze this training-history effect, first characterizing how forget-set influence accumulates along the optimization trajectory and then showing, in a two-layer linear model, that earlier introduction of the forget data leads to greater parameter displacement from the retain-only solution through shared feature directions. To test this hypothesis, we train language models from scratch while controlling when the forget set is introduced. Across different models, unlearning methods, and evaluation metrics, we find that knowledge maintained from earlier stages of pretraining is consistently more difficult to remove than the same knowledge introduced later. These results show that two models can fit the same forget data similarly yet differ substantially in how that knowledge is represented in their parameters and how readily it can subsequently be removed, depending on when the data entered training. Our findings motivate benchmarks that explicitly account for training history and the development of unlearning methods that are effective across more general and realistic training settings.

1 INTRODUCTION

The right to be forgotten, as embodied in regulations such as the GDPR, has motivated work on *machine unlearning*: removing the influence of particular training data from a learned model without retraining it from scratch (Cao & Yang, 2015; Bourtoule et al., 2021). This problem is especially consequential for large language models (LLMs), which can memorize sensitive information, copyrighted material, and other content that may later need to be removed (Carlini et al., 2021; Feldman, 2020). Machine-unlearning research therefore asks whether such knowledge can be removed efficiently while preserving the model's utility on the remaining data.

Standard LLM-unlearning evaluations typically begin with a pretrained model, introduce the forget set during a subsequent fine-tuning stage, and then evaluate whether an unlearning algorithm can remove it (Maini et al., 2024; Shi et al., 2024b). This protocol measures the removal of knowledge acquired near the end of the training pipeline. Deployment-time deletion requests, however, may concern data encountered during pretraining itself. The distinction is especially salient for LLM pretraining, where a single pass over a massive corpus is a practical training regime (Komatsuzaki, 2019): examples encountered at different points in the training stream undergo different amounts of subsequent optimization, with limited opportunity for later passes to erase this history.

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 dynamics provide reason to expect these acquisition histories to matter. Deep networks exhibit critical learning periods in which early perturbations to the data can have persistent effects on learned representations (Achille et al., 2017). More recently, Baek et al. (2026) show that exposure to domain-specific data during pretraining can produce qualitatively different behavior from introducing the same information through fine-tuning, while Feng et al. (2026) find that earlier exposure can make learned capabilities more persistent under subsequent adaptation. These findings motivate our central question: *Does success on fine-tuned forget sets transfer to knowledge acquired through pretraining when the model's current fit to the target data is comparable?*

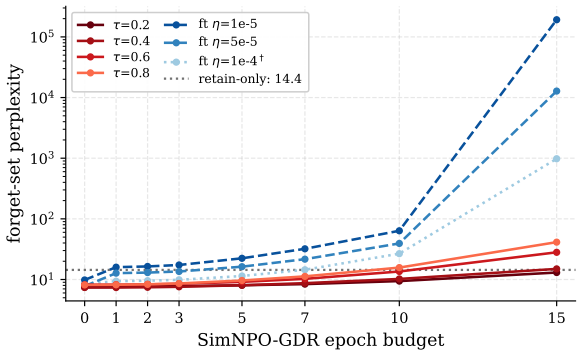

Figure 1: **Fine-tuned models are much more responsive to unlearning methods than the pretrained models.** All checkpoints begin at similar forget-set perplexity, but under the same epoch budget the fine-tuned controls (blue, dashed) rise much faster than the pretrained models (red, solid). Also a later encounter with the forget set (larger τ) leads to higher slope among the pretrained models.

Figure 1 previews our central result: unlearning responsiveness differs sharply across acquisition histories even when forget-set likelihood is matched. While the forget-set perplexity is comparable before unlearning, the same unlearning method would lead to better unlearning on the fine-tuned models (blue dashed lines). Also for the pretrained models (red solid lines) the earlier the forget-set is introduced (smaller τ), the worse the unlearning method works (see Section 5.2 for more details.

We hypothesize that acquisition history changes how target knowledge is integrated into model parameters. Data encountered while representations are developing can repeatedly shape features that later support other examples and capabilities. Introduced after those representations have formed, the same data may rely more on existing features, enabling a more localized intervention. This hypothesis identifies the complete acquisition history as the basis for unlearning response beyond pre-unlearning fit alone.

We first formalize this distinction theoretically (Section 3). For a general model trained by gradient descent, we show that exposure to forget data displaces the model from the retain-only trajectory through a sequence of forget–retain gradient discrepancies, each propagated by the subsequent retain-set dynamics (Proposition 3.2). We then analyze a two-layer linear network, where we prove that the parameter distance to the retrain model is non-increasing in the injection time of the forget set during training (Corollary 3.8). In this setting, Theorem 3.6 allows us to decompose the distance into terms associated with forget-set fit, the retain-loss gap, and features shared between the forget and retain data. This decomposition directly guides our experiments: we vary when the same forget data enters training while controlling forget and retain performance, preventing differences in endpoint fit from confounding the effect of training stage.

Our empirical results establish training history as an important dimension of machine unlearning. Across training stages, model families, unlearning methods, and evaluation criteria, knowledge maintained for longer during pretraining is substantially more resistant to removal than knowledge introduced later through pretraining or post-training fine-tuning. Earlier exposure also produces substantially larger displacement from the retain-only solution, consistent with the structural effect predicted by our theoretical results. Using common unlearning evaluation metrics we show that traces of earlier-acquired data persist longer under unlearning. Together, these results expose a limitation of existing benchmarks such as TOFU (Maini et al., 2024) and MUSE (Shi et al., 2024b): conclusions drawn from knowledge introduced only after pretraining need not transfer to knowledge introduced during pretraining, motivating unlearning evaluations that explicitly account for the training history of the data.

2 Related Work

Machine unlearning seeks to remove the influence of specified training data from an already-trained model. Benchmarks assess outcomes such as target recall, privacy leakage, and retained utility: TOFU introduces fictitious facts through fine-tuning, while MUSE evaluates removal of books and news articles (Maini et al., 2024; Shi et al., 2024b). These evaluations do not compare the same target data introduced at different points in training. Training dynamics suggest why that distinction matters. Individual examples can be learned and subsequently forgotten within a single training run (Toneva et al., 2018). Larger language models memorize examples faster and forget less during continued training (Tirumala et al., 2022), while verbatim memorization increases with model capacity and repeated exposure (Carlini et al., 2022). Factual knowledge accumulates across presentations and can be diluted by later updates (Chang et al., 2024). Continued training can reduce memorization of earlier examples (Jagielski et al., 2022), other examples undergo cycles of forgetting and relearning, with some early-memorized examples remaining accessible longer (Leybzon & Kervadec, 2024). Acquisition stage also shapes later behavior. Pretraining on specialized data can yield different outcomes from introducing it through fine-tuning (Baek et al., 2026), and early exposure to capability-relevant data can improve its retention during later fine-tuning (Feng et al., 2026). We examine the related question of targeted unlearning, comparing acquisition histories at similar initial forget-set perplexity across two model families and four unlearning methods. Studies of critical learning periods show that early training conditions can have persistent effects (Achille et al., 2017), while extended pretraining can reduce subsequent plasticity (Springer et al., 2025). Similarly, safety interventions introduced earlier in pretraining produce behaviors more robust to downstream fine-tuning (Sam et al., 2026). In a controlled Pythia study, Huang et al. (2024) find that later checkpoints memorize repeatedly injected sequences more readily; the tested unlearning methods often fail to remove those sequences without degrading utility. Yu et al. (2025) study path dependence in machine unlearning by varying when forget data is introduced across post-training stages starting from an already pretrained model. Interestingly, they observe an opposite recency trend in this post-training setting, with more recently introduced knowledge sometimes being harder to unlearn. This need not conflict with our findings: their experiments examine short post-training adaptations after the representation space has largely formed, whereas we vary forget-set exposure during *pretraining from scratch*, when the target data can directly shape representation formation and interact with subsequent learning. Thus, their results characterize path dependence in post-training optimization, while ours studies how acquisition history during representation formation affects later unlearning. Our work is also complementary to Feng et al. (2026), who show that earlier exposure can make learned capabilities more robust to later fine-tuning. Their work studies *retention under subsequent adaptation*, whereas we study *targeted removal through machine unlearning*. While their results suggest that early exposure can produce knowledge that is more resistant to overwriting, we ask whether early pretraining exposure also makes knowledge more deeply integrated with later-learned representations and therefore harder to selectively remove.

3 Methods

This section develops the theoretical foundation for temporal representational entanglement in two stages. After presenting the problem formulation in section 3.1, section 3.2 presents an architecture-agnostic result for models trained with gradient descent under local smoothness assumptions: when the forget set is mixed into training starting at step τT, where T is the total number of steps and $\tau \in [0, 1]$ (henceforth referred to as stage), the deviation from the retain-only trajectory is, up to a controlled remainder, a sum of $(1 - \tau)T$ forget–retain gradient discrepancies, each propagated through the subsequent retain dynamics. Earlier exposure therefore accumulates more such contributions. Section 3.3 specializes to a two-layer linear network that allows us to decompose the parameter distance from the retrain model to four criteria that guide the design of our experiments.

3.1 Problem Formulation

Let $\mathcal{D} = \mathcal{D}_r \cup \mathcal{D}_f$ denote the full pretraining corpus, partitioned into a *retain set* $\mathcal{D}_r$ and a *forget set* $\mathcal{D}_f$, where $|\mathcal{D}_f| \ll |\mathcal{D}_r|$. A language model f_θ is trained by minimizing the empirical cross-entropy loss: $\theta^* = \arg\min_\theta \mathcal{L}(\theta; \mathcal{D}_r \cup \mathcal{D}_f)$. Let T denote the total number of training iterations and let $\tau \in [0, 1]$ denote the *injection stage*: the fraction of training completed before any examples from

$\mathcal{D}_f$ are introduced. We define the stage-τ training procedure as: $\theta^*_\tau = \mathrm{Train}\Big(\mathcal{D}_r^{[0,\,T]},\ \mathcal{D}_f^{[\tau T,\,T]}\Big)$, where $\mathcal{D}_r^{[a,b]}$ denotes sampling from $\mathcal{D}_r$ during iterations $[a, b]$, and $\mathcal{D}_f^{[\tau T,T]}$ denotes that examples from $\mathcal{D}_f$ are mixed into the training stream, in a fixed proportion, from iteration τT until the end of training at iteration T. The standard benchmark protocol corresponds to $\tau \approx 1$: the forget set is introduced only at the very end of training via fine-tuning. Given a trained model θ^*_τ and a target forget set $\mathcal{D}_f$, an unlearning algorithm $\mathcal{U}$ produces: $\hat{\theta} = \mathcal{U}(\theta^*_\tau,\, \mathcal{D}_f,\, \mathcal{D}_r)$. The gold standard for unlearning is *retrain equivalence*: $\hat{\theta}$ should behave as if trained only on $\mathcal{D}_r$, i.e. $\hat{\theta} \approx \theta^*_r$, where θ^*_r denotes the model trained on $\mathcal{D}_r$ alone.

Definition 3.1 (Unlearning Cost)**.** *Define the* idealized retain-preserving unlearning cost

$$C(\tau) = \inf_{\Delta\theta} \|\Delta\theta\|_F \text{ s.t. } \begin{cases} \mathcal{L}(\theta^*_\tau + \Delta\theta; \mathcal{D}_f) \geq \mathcal{L}(\theta^*_r; \mathcal{D}_f) - \xi_f \\ \mathcal{L}(\theta^*_\tau + \Delta\theta; \mathcal{D}_r) \leq \mathcal{L}(\theta^*_r; \mathcal{D}_r) + \xi_r. \end{cases}$$

Here $\xi_f, \xi_r \geq 0$ are fixed tolerances: ξ_f bounds how far the unlearned model's forget loss may fall short of the retrain model's, and ξ_r the retain-loss degradation permitted relative to it.

This quantity characterizes the minimum parameter change required to achieve approximately retrain-equivalent forget and retain behavior, independent of any particular unlearning algorithm. Note that the retrain model is always feasible for this program: taking $\Delta\theta = \theta^*_r - \theta^*_\tau$ yields $\theta^*_\tau + \Delta\theta = \theta^*_r$, which satisfies both constraints for every $\xi_f, \xi_r \geq 0$, with equality when $\xi_f = \xi_r = 0$. Consequently $C(\tau) \leq \left\|\theta^*_\tau - \theta^*_r\right\|_F$, so the parameter distance is an *achievable* upper bound on the unlearning cost: retraining from scratch is one admissible way to meet the retain-preserving constraints, and $C(\tau)$ is the cost of the cheapest one. Enlarging either tolerance only enlarges the feasible set, so $C(\tau)$ is non-increasing in ξ_f and ξ_r and the zero-tolerance value is the largest.

Hypothesis. For a fixed unlearning algorithm $\mathcal{U}$ and evaluation budget, models with earlier exposure to $\mathcal{D}_f$ are less able to recover retrain-equivalent behavior:

$$\mathrm{dist}(\mathcal{U}(\theta^*_\tau,\, \mathcal{D}_f,\, \mathcal{D}_r),\ \theta^*_r) \ \uparrow \text{ as } \tau \downarrow . \tag{1}$$

That is, under the same unlearning procedure and budget, models trained with earlier exposure to $\mathcal{D}_f$ remain farther from the retrain-from-scratch baseline.

3.2 General Optimization Perspective

We begin with a general differentiable model parameterized by $\theta \in \mathbb{R}^p$. Let $L_r(\theta) = \mathcal{L}(\theta; \mathcal{D}_r)$ and $L_f(\theta) = \mathcal{L}(\theta; \mathcal{D}_f)$ denote the retain and forget losses. We define $\tau \in [0, 1]$ as the stage at which the forget set is introduced during training: for the first τT steps the model is trained on $\mathcal{D}_r$ alone, and afterwards on the mixture $(1-\gamma)\mathcal{D}_r + \gamma\mathcal{D}_f$, i.e. on $(1-\gamma)L_r + \gamma L_f$, with $\gamma \in (0, 1]$ fixed. Let:

- θ^*_τ denote the final parameters obtained when forget exposure begins at stage τ,
- θ^*_r denote the retain-only model trained exclusively on $\mathcal{D}_r$ ($\gamma = 0$).

Our goal is to understand how the timing of forget-set exposure affects the difficulty of recovering retrain-equivalent behavior. The following proposition shows that forget information introduced earlier in training propagates through a larger portion of the optimization trajectory and increasingly influences subsequent retained-task optimization.

Proposition 3.2. *Assume $L_r(\theta)$ is twice continuously differentiable and its Hessian is locally ρ-Lipschitz along the training trajectory. Assume also that the retain-loss Hessian is locally bounded as $0 \preceq \nabla^2 L_r(\theta^r_t) \preceq \lambda I$, and gradient descent uses a learning rate $0 < \eta \leq \frac{1}{\lambda}$. Let θ^r_t denote the retain-only trajectory and θ_t denote the trajectory trained on the mixture from step τT. Define $\delta_t = \theta_t - \theta^r_t, H_t = \nabla^2 L_r(\theta^r_t), g^{(t)}_r = \nabla L_r(\theta^r_t), g^{(t)}_f = \nabla L_f(\theta^r_t)$. Then the final deviation satisfies*

$$\theta^*_\tau - \theta^*_r = -\eta\gamma \sum_{s=\tau T}^{T-1} \left(\prod_{t=s+1}^{T-1} \big(I - \eta(1-\gamma)H_t\big) \right) \big(g^{(s)}_f - g^{(s)}_r\big) + R_T,$$

where the approximation error obeys

$$\|R_T\| \leq \eta \sum_{s=\tau T}^{T-1} \Big((1-\gamma)\frac{\rho}{2}\|\delta_s\|^2 + \gamma\|\nabla L_f(\theta_s) - \nabla L_f(\theta^r_s)\| \Big).$$

If, additionally, ∇L_f is locally M_f-Lipschitz and $\|\delta_t\| \le \delta_{\max}$ along the trajectory, then

$$\|R_T\| \le \eta(1-\tau)T\left((1-\gamma)\frac{\rho}{2}\delta_{\max}^2 + \gamma M_f \delta_{\max}\right).$$

Proposition 3.2 shows that forget information introduced earlier influences a larger fraction of subsequent optimization updates, increasing the opportunity for interaction with later learned representations. The driving term is the gradient discrepancy $g_f^{(s)} - g_r^{(s)}$ scaled by the mixture weight γ: directions in which the two losses agree contribute nothing, while each step of forget exposure injects a deviation that is then propagated by the retain dynamics $I - \eta(1-\gamma)H_t$. Importantly, this result does not rely on a specific architecture and applies broadly to differentiable models trained with gradient descent under local smoothness assumptions.

3.3 Two-Layer Linear Networks

We follow the setup of Feng et al. (2026) in this section; we use a two-layer linear network, where feature dynamics decouple exactly and follow their mixture optimization scheme after introducing the forget set. More specifically, consider

$$f_\theta(x) = W_2 W_1 x, \quad \theta = (W_1, W_2), \quad W_1, W_2 \in \mathbb{R}^{d\times d},$$

trained by gradient descent on the squared loss (in this section $\mathcal{L}$ denotes this loss rather than the cross-entropy) $\mathcal{L}(\theta;\mathcal{D}) = \mathbb{E}_{x\sim\mathcal{D}}\|W_2W_1x - A_{\mathcal{D}}x\|^2$, where $A_{\mathcal{D}} \in \{A_r, A_f\}$ is the ground-truth linear map of dataset $\mathcal{D} \in \{\mathcal{D}_r, \mathcal{D}_f\}$, and $\mathbb{E}_{x\sim\mathcal{D}}[xx^\top] = I_d$. Training follows the stage-controlled protocol of Section 3.2.

Assumption 3.3. *There exist orthonormal $U, V \in \mathbb{R}^{d\times d}$ such that $A_r = U\Sigma_r V^\top$ and $A_f = U\Sigma_f V^\top$, with Σ_r, Σ_f diagonal and non-negative.*

This is Assumption C.5 of Feng et al. (2026) and allows singular-value dynamics to decouple across feature directions.

Feature decomposition. We partition the d feature directions into four blocks:

- **Invariant** ($i \in \mathcal{I}$): $(\Sigma_r)_{ii} = (\Sigma_f)_{ii} = \sigma_i^{\text{inv}} > 0$. Equally important to both tasks.
- **Retain-specific** ($i \in \mathcal{R}$): $(\Sigma_r)_{ii} = \sigma_i^r > 0$, $(\Sigma_f)_{ii} = 0$. Relevant only to $\mathcal{D}_r$.
- **Forget-specific** ($i \in \mathcal{F}$, $|\mathcal{F}| = k$): $(\Sigma_f)_{ii} = \beta > 0$, $(\Sigma_r)_{ii} = 0$. Relevant only to $\mathcal{D}_f$.
- **Common** ($i \in \mathcal{C}$, $|\mathcal{C}| = k_c$): $(\Sigma_f)_{ii} = \sigma_i^f > 0$ and $(\Sigma_r)_{ii} = \sigma_i^c > 0$, but $\sigma_i^c < \sigma_i^f$. These features appear in $\mathcal{D}_f$ with strong, coherent signal σ_i^f and in $\mathcal{D}_r$ with weak, diffuse signal σ_i^c: only a sparse fraction $p_c \ll 1$ of $\mathcal{D}_r$ samples (i.e. $n_c = p_c|\mathcal{D}_r|$ samples) carry these features, so $\sigma_i^c = p_c \cdot \tilde{\sigma}_i^c$ with $\tilde{\sigma}_i^c$ the local signal strength and p_c the coverage fraction.

The forget-specific block captures features unique to $\mathcal{D}_f$ (e.g. character names, specific plot details). The common block captures features present in $\mathcal{D}_f$ coherently that also appear sporadically in $\mathcal{D}_r$ (e.g. narrative or linguistic patterns shared with a small fraction of the general corpus).

Assumption 3.4. *$W_1(0) = \varepsilon V^\top$ and $W_2(0) = \varepsilon U$ with $\varepsilon = e^{-T_0}$ and $\varepsilon^2 < \min\{(\Sigma_r)_{ii}, (\Sigma_f)_{ii} : \text{entry} > 0\}$. The step size satisfies $6\eta S < 1$, where $S := \max_i \max\{(\Sigma_r)_{ii}, (\Sigma_f)_{ii}\}$.*

This mirrors Assumption C.8 of Feng et al. (2026) and the vanishing-initialization regime of Gidel et al. (2019): the network starts small-but-nonzero, which is what makes features emerge sequentially. Additionally, the step size is bounded as Assumption C.12 of Feng et al. (2026), in terms of the largest target singular value.

Assumption 3.5. *$2\eta\,\sigma_{\max}^c T \le \kappa T_0$ for some $\kappa \in [0,1)$, where $\sigma_{\max}^c = \max_{i\in\mathcal{C}} \sigma_i^c = p_c \max_{i\in\mathcal{C}} \tilde{\sigma}_i^c$.*

Assumption 3.5 states that the sparse retain-set signal alone is too weak for common features to escape the small initialization, so the retrain model does not learn them.

Theorem 3.6. *Under Assumptions 3.3–3.5, let $\ell_f := \ell_f(\theta^*_\tau)$ and $\Delta\ell_r := \mathcal{L}(\theta^*_\tau; \mathcal{D}_r) - \mathcal{L}(\theta^*_r; \mathcal{D}_r)$. Then for every $\tau \in [0, 1)$,*

$$\left\|\theta^*_\tau - \theta^*_r\right\|_F^2 \;\geq\; \frac{1}{2}\Bigg[\underbrace{2k\Big(\sqrt{\big(\beta - \sqrt{\ell_f/k}\big)_+} - \varepsilon\Big)_+^2}_{\textit{forget fit}} \;+\; \underbrace{\frac{(\Delta\ell_r)_+^2}{8\,d\,S^3}}_{\textit{retain fit}} \;+\; \underbrace{2\sum_{i\in\mathcal{C}}\Big(G_i(\tau) - \varepsilon^{1-\kappa}\Big)_+^2}_{\textit{common features}}\Bigg], \quad (2)$$

where $s_i := (1-\gamma)\sigma_i^c + \gamma\sigma_i^f$ and

$$G_i(\tau) = \min\Big\{\sqrt{s_i/2},\; \varepsilon\big(1 + \eta\, s_i\big)^{(1-\tau)T}\Big\}.$$

The forget-fit term is non-increasing in ℓ_f, and the retain-fit term is non-decreasing in $\Delta\ell_r$. The common-feature term does not depend on ℓ_f or $\Delta\ell_r$. It is non-increasing in τ and non-decreasing in p_c (hence in n_c) and in γ. It is a sum of k_c non-negative terms, so it grows with the number of common features ($2k_c(G(\tau) - \varepsilon^{1-\kappa})_+^2$ when all common features share the same strengths).

All proofs are deferred to Appendix A.2.

Intuition. The bound separates the four quantities that govern how far a stage-τ model sits from the retrain model.

(i) *Injection stage.* Through the exponent $(1-\tau)T$, earlier injection gives common features more steps of joint reinforcement by $\mathcal{D}_f$ and the n_c retain samples that carry them. The retrain model never learns these features (Assumption 3.5).
(ii) *Data Entanglement.* The common-feature term grows with the fraction p_c of retain samples sharing features with the forget set and with the number k_c of shared features.
(iii) *Forget loss.* The forget-fit term depends on the forget-specific features only through ℓ_f: a model that fits $\mathcal{D}_f$ better must have moved further along $\mathcal{F}$.
(iv) *Retain loss.* The retain-fit term charges the displacement revealed by a retain-loss gap to the retrain model.

Item (ii) depends only on the data and is therefore fixed across our models, which share $\mathcal{D}_r$ and $\mathcal{D}_f$. In the two-layer analysis, holding the mixture weight and endpoint losses fixed isolates the bound's dependence on injection stage. Our empirical protocol instead uses repeated, perplexity-gated injections to obtain comparable initial forget-set fit; its cumulative exposure varies across stages.

Remark 3.7. *In Theorem 3.6, two models with the same forget loss ℓ_f and retain-loss gap $\Delta\ell_r$ have identical forget-fit and retain-fit terms in the lower bound. However, their common-feature terms depend explicitly on the injection stage. Thus, even after matching forget and retain losses, the theoretical bound retains a dependence on the training history of the forget set.*

Key Observation: *Perplexity measures how well a dataset is fit, but not how that fit is represented in the learned parameters.*

Corollary 3.8 shows that the actual parameter distance $\|\theta^*_\tau - \theta^*_r\|_F$ is non-increasing in τ. Consequently, models with comparable endpoint losses can differ in their displacement from the retain-only solution as a consequence of *when* the forget data was introduced.

Corollary 3.8. *Under Assumptions 3.3-3.4 and a fixed mixture weight $\gamma \in (0, 1]$, let θ^*_τ denote the model obtained when the forget set is introduced at stage τ, and let θ^*_r denote the retain-only model. For any $0 \leq \tau_1 < \tau_2 < 1$ (with $\tau_1 T$ and $\tau_2 T$ integer),*

$$\left\|\theta^*_{\tau_1} - \theta^*_r\right\|_F \geq \left\|\theta^*_{\tau_2} - \theta^*_r\right\|_F.$$

Thus, within the two-layer linear model, introducing the forget set earlier cannot decrease the final parameter displacement from the retain-only trajectory.

4 Experimental Setup

Our experiments require training LLMs from scratch for the retain-only model and from different stages of pretraining for each value of τ. To evaluate whether the observed effects generalize across

architectures and scales, we use two models from different model families and of different sizes: Pythia-410M (Biderman et al., 2023) and TinyLlama-1.1B (Zhang et al., 2024). For training these models from scratch, we use the Pile (Gao et al., 2020) dataset. The forget corpus $\mathcal{D}_f$ is 411 passages (405 unique) from *The House in the Cerulean Sea* (HiCS): 570 KB of raw text, 142,869 GPT-NeoX BPE tokens. During injection the model is trained on $\mathcal{D}_f$ together with passages from the same fictional universe but disjoint from $\mathcal{D}_f$ (1.76 MB combined, 424,201 GPT-NeoX tokens). This setting matches the setting of the common unlearning benchmark MUSE Shi et al. (2024b), which uses Harry Potter text books and its fan-fictions. HiCS was chosen because based on its publishing date, we know it does not appear in the Pile dataset. Training is conducted on 4 NVIDIA H100 80GB GPUs. For more details on the setup of the experiments see Appendix B.

5 Results

In this section, we answer two main questions: (1) Do the theoretical results hold empirically? We test this first on synthetic data constructed to satisfy the assumptions of Theorem 3.6, and then on Pythia-410M and TinyLlama-1.1B, where we ask whether earlier injection of the forget set leaves the model further from the retrain solution (Section 5.1 and Appendix C.1). (2) Does earlier injection make the forget set harder to remove? We measure this under a fixed unlearning budget, through forget-set perplexity (Section 5.2) and membership inference under three different methods (Section 5.3).

5.1 Earlier Injection and Parameter Distance

Theorem 3.6 gives a history-dependent lower bound on parameter distance, while Corollary 3.8 proves monotonic distance under a fixed mixture weight in the two-layer linear model. Appendix C.1 examines the ordering in synthetic data that satisfies the assumptions in Section 3.3. With the forget loss matched across arms by bisection, the squared distance to the retain-only solution decreases strictly in τ ($11.59 \rightarrow 10.46$), and the change is confined to the common-feature block, where Theorem 3.6 places it. The bound equation 2 holds in all 2,025 random configurations we sample, and sweeps confirm the effect depends on shared features with the retain dataset.

To evaluate this in practice, we measure that distance directly in Pythia-410m, reporting $\|\theta - \theta_{\text{retain}}\|_2$ over all parameters for various pretrained and fine-tuned models. We first fine-tune the retrain model on $\mathcal{D}_f$, sweeping the learning rate from 10^{-6} to 2×10^{-3}, as shown in Figure 2(b) to find the setting that achieves the lowest perplexity on $\mathcal{D}_f$ (around 8) without compromising the utility of the model on the retain set (10.6–16.0 vs. 10.5 in the retrain model). For $\tau \in \{0.2, 0.4, 0.6, 0.8\}$, $\mathcal{D}_f$ is injected at step τT and its perplexity is then controlled for the remaining $(1-\tau)T$ steps to be comparable to the chosen fine-tuned models. So τ varies the duration over which the forget set shapes the weights at a comparable fit.

Figure 2(a) shows that parameter distance decreases strictly with τ, consistent with the ordering proved for the two-layer linear model in Corollary 3.8. Because the models have comparable ℓ_f, this ordering is consistent with the common-feature term in Eq. 2, which is non-increasing in τ. For the fine-tuned models, Figure 2(a) shows that reaching even the $\tau = 0.8$ line takes an η between 5×10^{-4} and 10^{-3}. At that scale the model is destroyed rather than better fitted as shown in Figure 2(b) (forget perplexity 28.2, WikiText-2 perplexity 1.7×10^3).

Figure 6 shows the same monotonic trend holds for TinyLlama-1.1B models as well and the results are not specific to a single family of models (see Section C.2 for details). Together, these results show that forget-set perplexity measures how well $\mathcal{D}_f$ is fit, but not which features carry that fit or how far training has moved from the retain-only endpoint. Sections 5.2 and 5.3 test whether this structural displacement corresponds to greater unlearning difficulty.

5.2 Forget-Set Response Across Unlearning Methods

Section 5.1 established that earlier injection leads to farther distance from θ_{retain} when $\mathcal{D}_f$ is comparable. We now ask what that displacement costs an unlearning method. We first compare how forget-set perplexity changes when the same unlearning method is applied to pretrained and fine-tuned checkpoints. Each trajectory begins at the corresponding pre-unlearning value, shown at epoch

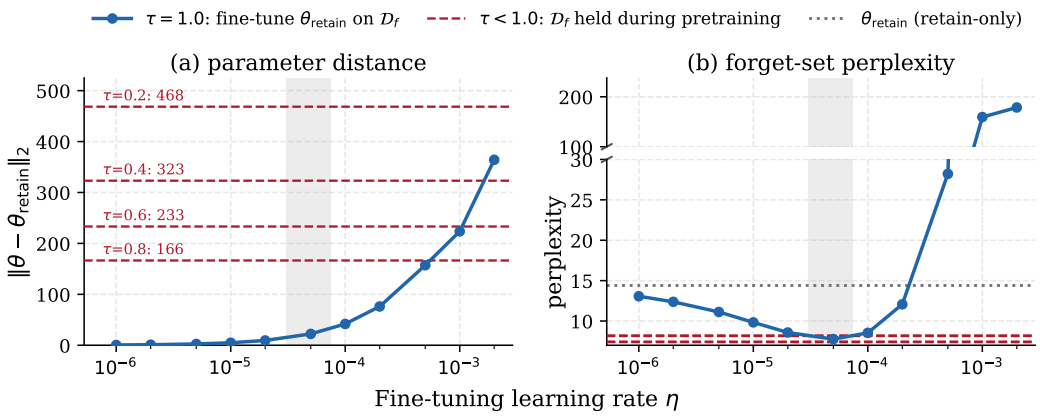

Figure 2: **Parameter distance to the retrain model is strictly non-increasing in τ.** Each panel shows the fine-tuned models ($\tau = 1.0$) as a function of learning rate η. The shaded region marks the fine-tuned models whose forget-set perplexity matches the target used for the pretrained models. Dashed lines show the four pretrained models at $\tau \in \{0.2, 0.4, 0.6, 0.8\}$; the dotted line is θ_{retain}. **(a)** Distance from θ_{retain}. **(b)** Forget-set perplexity. The pretrained models have similar perplexity by design, while the fine-tuning curve is non-monotone.

zero, and uses the same unlearning hyperparameters across training histories. Thus, the figures display both the initial forget-set fit and the subsequent response to unlearning.

Figure 1 shows that, under SimNPO-GDR, fifteen epochs increase forget perplexity by $5.1\times$, $3.4\times$, $2.7\times$, and $2.2\times$ for $\tau = 0.8, 0.6, 0.4, 0.2$, respectively, compared with $1648\times$ for the closest-matched fine-tuned model ($\eta = 5 \times 10^{-5}$; initial forget set perplexity of 7.76 versus 7.39–8.20). The response is monotone with maintenance duration: the $\tau = 0.2$ arm ends at 16.67, while the fine-tuned control exceeds 10^4.

Table 1: **Forget-set perplexity after unlearning depends strongly on training history.** The Initial column reports forget set perplexity before unlearning, and each method column reports this perplexity after its fixed evaluation budget. The retain-only baseline has a perplexity of 14.40 on $\mathcal{D}_f$. For fine-tuned checkpoints, η_{train} is the learning rate used to introduce HiCS, not the unlearning learning rate.

Regime	Setting	Initial	GA_{gdr}	SimNPO	RMU	CE-U
Pretrained	$\tau = 0.2$	7.45	19.66	16.67	45.38	198.36
Pretrained	$\tau = 0.4$	7.39	52.12	19.88	63.29	289.93
Pretrained	$\tau = 0.6$	8.20	416.73	28.07	73.12	644.46
Pretrained	$\tau = 0.8$	8.14	1091	41.33	78.41	825.32
Fine-tuned	$\eta_{\text{train}} = 10^{-5}$	9.83	6.0e7	1.91e5	241.62	1263.98
Fine-tuned	$\eta_{\text{train}} = 5{\times}10^{-5}$	7.77	2.9e13	1.28e4	225.28	1472.06
Fine-tuned	$\eta_{\text{train}} = 10^{-4}$	8.52	2.6e18	980	415.19	4379.02

Table 1 shows the results for other unlearning methods. Rows enumerate the pretrained checkpoints at different values of τ and the fine-tuned checkpoints obtained with different learning rates of finetuning. For each model within a row, the initial forget set perplexity is reported followed by the post-unlearning value for each unlearning method. For Pythia-410M, fine-tuned models consistently exhibit larger increases in forget set perplexity than pretrained models. More importantly, among the pretrained models, post-unlearning forget set perplexity increases monotonically with τ under all four unlearning methods.

Appendix C presents similar results for TinyLlama-1.1B; Figure 7 shows the epoch-by-epoch SimNPO-GDR response, and Table 2 reports the overall results for other unlearning methods. Together, they show similar trends in effectiveness of unlearning methods in removing the forget set from the model (i.e., increasing its perplexity). These results show that the initial fit to the forget

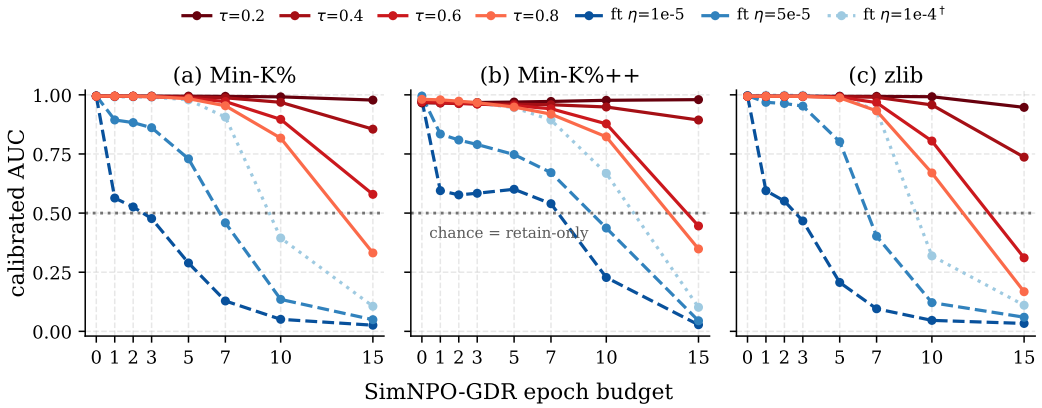

Figure 3: **Calibrated membership-inference AUC, forget set against unseen samples**, under three methods. Calibration subtracts the retain-only model's score, so 0.5 is both chance and exactly where θ_{retain} sits. Models easier to unlearn (e.g., fine-tuned models) reach the chance level in fewer epochs whereas models harder to unlearn (e.g., $\tau = 0.2$ may barely fall below an AUC of 1 even after 15 epochs.

set (perplexity) does not reliably predict the effectiveness of unlearning, making training history an important dimension of unlearning evaluation.

5.3 Membership Inference

Membership Inference Attacks (MIAs) are commonly used to evaluate the effectiveness of unlearning methods Shi et al. (2024b). These attacks are utilized to predict whether the forget samples are separable from the unseen samples. Figure 3 reports the results for three different MIAs (see Section B for more details on these attacks). The AUC is calibrated for each of the three scoring rules such that the retrain model (the model used for fine-tuning) achieves an AUC of 0.5 implying no better than random guess for detecting the forget samples. As the figure shows, for all the MIAs, the ordering is preserved with smaller values of τ being more difficult to unlearn. For the $\tau = 0.2$ model, the curve barely moves away from a perfect AUC even after 15 epochs, whereas for the fine-tuned models often a few steps of unlearning is enough to reach the chance level (AUC of 0.5) and all the curves for fine-tuned models reach near zero values after 15 epochs (over unlearning). This observation is consistent under the same attacks for TinyLlama-1.1B and also other unlearning methods applied to Pythia models (see Appendix C.4 for more results).

6 Conclusion

We show that the training history of a forget set can substantially affect how difficult it is to remove, even when models have comparable pre-unlearning fit to that data. Both our theoretical analysis and experiments suggest that earlier exposure during pretraining is associated with greater integration with representations subsequently reused by the retain data, making the same knowledge less responsive to standard unlearning methods. These findings highlight a limitation of current benchmarks that evaluate primarily post-training or fine-tuned knowledge and motivate unlearning evaluations that explicitly account for when and how target data entered training.

Our study has several limitations. The theoretical monotonicity result is established in a simplified two-layer linear setting, while the proposed shared-feature mechanism is only indirectly supported in transformer experiments. Empirically, we study two model families and a single primary forget corpus, and our main comparisons use fixed unlearning budgets rather than independently optimized forget–retain tradeoffs for each training history. Future work should test whether the effect generalizes across broader classes of data and models, directly measure the representational mechanisms underlying it, and characterize the best achievable unlearning performance across different acquisition histories.

AI USE STATEMENT

We used generative AI tools to improve the clarity, grammar, and readability of the manuscript and to assist with writing and editing portions of the research code. The authors reviewed and tested the AI-assisted code, checked the edited text for technical accuracy, and take responsibility for the final content of the paper and its artifacts.

ETHICS STATEMENT

This study uses passages from *The House in the Cerulean Sea* as a controlled forget corpus for evaluating machine unlearning. Releasing source passages or model-generated excerpts could reproduce material from that work. We report aggregate experimental results and will release text or model artifacts only after reviewing their provenance, redistribution rights, and risk of substantial verbatim reproduction.

REPRODUCIBILITY STATEMENT

We have taken several steps to make our results reproducible. All models are built from public artifacts. Our 20k-step Pythia-410M runs use the Pythia training configuration and read the Pile in Pythia's fixed data order, so every retain-only checkpoint and every injection branch point can be regenerated exactly. The full-training replication uses the released Pythia checkpoints directly. The injection protocol, including the perplexity gate used to build the depth-matched ("enforced") checkpoints, is specified in Section 5 and Appendix B.5. Training hyperparameters and random seeds are listed in Appendix B. For each unlearning method, Appendix B.1 reports the hyperparameter search space, the selected configuration and the selection criterion. The assumptions behind Proposition 3.2 are stated in full, with complete proofs, in Appendix A. The supplementary material contains anonymized code for pretraining, injection, unlearning and evaluation, along with configuration files and the scripts that produce every figure and table. The forget corpus is a copyrighted novel, so we do not redistribute its text. Instead, the supplement includes the preprocessing and chunking scripts and per-chunk hashes, so that $\mathcal{D}_f$ and the probe items can be rebuilt from a legally obtained copy.

REFERENCES

Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep neural networks. *arXiv preprint arXiv:1711.08856*, 2017.

Christina Baek, Ricardo Pio Monti, David Schwab, Amro Abbas, Rishabh Adiga, Cody Blakeney, Maximilian Böther, Paul Burstein, Aldo Gael Carranza, Alvin Deng, et al. The finetuner's fallacy: When to pretrain with your finetuning data. *arXiv preprint arXiv:2603.16177*, 2026.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International conference on machine learning*, pp. 2397–2430. PMLR, 2023.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. Gpt-neox-20b: An open-source autoregressive language model. In *Proceedings of BigScience Episode# 5–Workshop on Challenges & Perspectives in Creating Large Language Models*, pp. 95–136, 2022.

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

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pp. 463–480. IEEE, 2015.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650. USENIX Association, August 2021. ISBN 978-1-939133-24-3. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. How do large language models acquire factual knowledge during pretraining? *Advances in neural information processing systems*, 37:60626–60668, 2024.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *Advances in Neural Information Processing Systems*, 38:1540–1567, 2025.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd annual ACM SIGACT symposium on theory of computing*, pp. 954–959, 2020.

Lawrence Feng, Gaurav R Ghosal, Jacob Mitchell Springer, Ziqian Zhong, and Aditi Raghunathan. Early data exposure improves robustness to subsequent fine-tuning. *arXiv preprint arXiv:2605.12705*, 2026.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Jing Huang, Diyi Yang, and Christopher Potts. Demystifying verbatim memorization in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10711–10732, 2024.

Matthew Jagielski, Om Thakkar, Florian Tramer, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, et al. Measuring forgetting of memorized training examples. *arXiv preprint arXiv:2207.00099*, 2022.

Aran Komatsuzaki. One epoch is all you need. *arXiv preprint arXiv:1906.06669*, 2019.

Danny D Leybzon and Corentin Kervadec. Learning, forgetting, remembering: Insights from tracking llm memorization during training. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 43–57, 2024.

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.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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.

Dylan Sam, Sachin Goyal, Pratyush Maini, Alexander Robey, and J Zico Kolter. When should we introduce safety interventions during pretraining? *arXiv preprint arXiv:2601.07087*, 2026.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=zWqr3MQuNs.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. *arXiv preprint arXiv:2407.06460*, 2024b.

Jacob Mitchell Springer, Sachin Goyal, Kaiyue Wen, Tanishq Kumar, Xiang Yue, Sadhika Malladi, Graham Neubig, and Aditi Raghunathan. Overtrained language models are harder to fine-tune. *arXiv preprint arXiv:2503.19206*, 2025.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

Bo Yang. Ce-u: Cross entropy unlearning. *arXiv preprint arXiv:2503.01224*, 2025.

Jiatong Yu, Yinghui He, Anirudh Goyal, and Sanjeev Arora. On the impossibility of retrain equivalence in machine unlearning. *arXiv preprint arXiv:2510.16629*, 2025.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Yang, and Hai Li. Min-k%++: Improved baseline for pre-training data detection from large language models. In *International Conference on Learning Representations*, volume 2025, pp. 64845–64862, 2025.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

A PROOFS

In this appendix, we provide detailed proofs for the theoretical results presented in Section 3. We begin with the general optimization-based analysis and then specialize to the representation-learning setting based on deep linear networks.

A.1 PROOF OF PROPOSITION 3.2

Proof. Let θ_t^r denote the retain-only trajectory:

$$\theta_{t+1}^r = \theta_t^r - \eta \nabla L_r(\theta_t^r).$$

Both trajectories start from the same initialization and coincide for $t < \tau T$. After the forget set is introduced at step τT, the forget-exposed trajectory performs gradient descent on the mixture loss $(1-\gamma)L_r + \gamma L_f$:

$$\theta_{t+1} = \theta_t - \eta(1-\gamma)\nabla L_r(\theta_t) - \eta\gamma \nabla L_f(\theta_t).$$

Define the trajectory deviation

$$\delta_t = \theta_t - \theta_t^r.$$

Writing the retain-only step as $\nabla L_r(\theta_t^r) = (1-\gamma)\nabla L_r(\theta_t^r) + \gamma \nabla L_r(\theta_t^r)$ and subtracting it from the forget-exposed update gives

$$\delta_{t+1} = \delta_t - \eta(1-\gamma)\left(\nabla L_r(\theta_t) - \nabla L_r(\theta_t^r)\right) - \eta\gamma\left(\nabla L_f(\theta_t) - \nabla L_r(\theta_t^r)\right).$$

By Taylor's theorem around θ_t^r,

$$\nabla L_r(\theta_t) = \nabla L_r(\theta_t^r) + H_t \delta_t + r_t,$$

where $H_t = \nabla^2 L_r(\theta_t^r)$. Since the Hessian of L_r is locally ρ-Lipschitz, we have

$$\|\nabla^2 L_r(\theta) - \nabla^2 L_r(\theta')\| \le \rho \|\theta - \theta'\|$$

for all θ, θ' along the local training trajectory. Applying Taylor's theorem with integral remainder around θ_t^r, we obtain

$$\nabla L_r(\theta_t) = \nabla L_r(\theta_t^r) + \nabla^2 L_r(\theta_t^r)(\theta_t - \theta_t^r) + r_t,$$

where the remainder term is

$$r_t = \int_0^1 \left(\nabla^2 L_r(\theta_t^r + \alpha\delta_t) - \nabla^2 L_r(\theta_t^r)\right) \delta_t \, d\alpha,$$

and $\delta_t = \theta_t - \theta_t^r$. Taking norms and applying the Lipschitz Hessian assumption,

$$\|r_t\| \le \int_0^1 \left\|\nabla^2 L_r(\theta_t^r + \alpha\delta_t) - \nabla^2 L_r(\theta_t^r)\right\| \|\delta_t\| \, d\alpha.$$

Using

$$\left\|\nabla^2 L_r(\theta_t^r + \alpha\delta_t) - \nabla^2 L_r(\theta_t^r)\right\| \le \rho\alpha\|\delta_t\|,$$

we obtain

$$\|r_t\| \le \int_0^1 \rho\alpha\|\delta_t\|^2 \, d\alpha.$$

Evaluating the integral gives

$$\|r_t\| \le \rho\|\delta_t\|^2 \int_0^1 \alpha d\alpha = \frac{\rho}{2}\|\delta_t\|^2.$$

Therefore, the Taylor remainder is quadratically bounded as

$$\|r_t\| \le \frac{\rho}{2}\|\delta_t\|^2.$$

Next, write the forget gradient as

$$\nabla L_f(\theta_t) = \nabla L_f(\theta_t^r) + q_t,$$

where $q_t = \nabla L_f(\theta_t) - \nabla L_f(\theta_t^r)$. Using the notation $g_r^{(t)} = \nabla L_r(\theta_t^r)$ and $g_f^{(t)} = \nabla L_f(\theta_t^r)$, we obtain

$$\nabla L_f(\theta_t) - \nabla L_r(\theta_t^r) = g_f^{(t)} - g_r^{(t)} + q_t.$$

Substituting these two decompositions into the deviation recursion gives

$$\delta_{t+1} = \big(I - \eta(1-\gamma)H_t\big)\delta_t - \eta\gamma\big(g_f^{(t)} - g_r^{(t)}\big) - \eta\big((1-\gamma)r_t + \gamma q_t\big).$$

Let $A_t = I - \eta(1-\gamma)H_t$. Then

$$\delta_{t+1} = A_t\delta_t - \eta\gamma\big(g_f^{(t)} - g_r^{(t)}\big) - \eta\big((1-\gamma)r_t + \gamma q_t\big).$$

At the injection time, the two trajectories coincide, so $\delta_{\tau T} = 0$. Unrolling the recurrence from $t = \tau T$ to $T-1$ gives

$$\delta_T = -\eta\gamma \sum_{s=\tau T}^{T-1} \left(\prod_{t=s+1}^{T-1} A_t\right) \big(g_f^{(s)} - g_r^{(s)}\big) - \eta \sum_{s=\tau T}^{T-1} \left(\prod_{t=s+1}^{T-1} A_t\right) \big((1-\gamma)r_s + \gamma q_s\big).$$

Substituting back $A_t = I - \eta(1-\gamma)H_t$, we get

$$\delta_T = -\eta\gamma \sum_{s=\tau T}^{T-1} \left(\prod_{t=s+1}^{T-1} \big(I - \eta(1-\gamma)H_t\big)\right) \big(g_f^{(s)} - g_r^{(s)}\big) + R_T,$$

where

$$R_T = -\eta \sum_{s=\tau T}^{T-1} \left(\prod_{t=s+1}^{T-1} \big(I - \eta(1-\gamma)H_t\big)\right) \big((1-\gamma)r_s + \gamma q_s\big).$$

Since $\delta_T = \theta_\tau^* - \theta_r^*$, this is the claimed decomposition.

It remains to bound R_T. Since

$$0 \preceq H_t \preceq \lambda I, \qquad 0 < \eta \leq \frac{1}{\lambda}, \qquad \gamma \in (0, 1],$$

we have $0 \preceq \eta(1-\gamma)H_t \preceq (1-\gamma)I \preceq I$, so all eigenvalues of $I - \eta(1-\gamma)H_t$ lie in $[\gamma, 1] \subseteq [0, 1]$. Therefore,

$$\|I - \eta(1-\gamma)H_t\| \leq 1.$$

Hence,

$$\left\| \prod_{t=s+1}^{T-1} \big(I - \eta(1-\gamma)H_t\big) \right\| \leq 1.$$

Using this in the definition of R_T, we obtain

$$\|R_T\| \leq \eta \sum_{s=\tau T}^{T-1} \|(1-\gamma)r_s + \gamma q_s\|.$$

By the triangle inequality,

$$\|R_T\| \leq \eta \sum_{s=\tau T}^{T-1} \left((1-\gamma)\|r_s\| + \gamma\|q_s\|\right).$$

Using the Taylor remainder bound,

$$\|r_s\| \leq \frac{\rho}{2}\|\delta_s\|^2,$$

and the definition of q_s,

$$\|q_s\| = \|\nabla L_f(\theta_s) - \nabla L_f(\theta_s^r)\|,$$

we obtain

$$\|R_T\| \leq \eta \sum_{s=\tau T}^{T-1} \left((1-\gamma)\frac{\rho}{2}\|\delta_s\|^2 + \gamma\|\nabla L_f(\theta_s) - \nabla L_f(\theta_s^r)\|\right).$$

This proves the first error bound. Now suppose additionally that ∇L_f is locally M_f-Lipschitz and that $\|\delta_t\| \leq \delta_{\max}$ along the trajectory. Then

$$\|\nabla L_f(\theta_s) - \nabla L_f(\theta_s^r)\| \leq M_f\|\theta_s - \theta_s^r\| = M_f\|\delta_s\| \leq M_f\delta_{\max}.$$

Also $\|\delta_s\|^2 \leq \delta_{\max}^2$. Therefore,

$$\|R_T\| \leq \eta \sum_{s=\tau T}^{T-1} \left((1-\gamma)\frac{\rho}{2}\delta_{\max}^2 + \gamma M_f\delta_{\max}\right).$$

There are $(1-\tau)T$ terms in the sum, so

$$\|R_T\| \leq \eta(1-\tau)T \left((1-\gamma)\frac{\rho}{2}\delta_{\max}^2 + \gamma M_f\delta_{\max}\right).$$

This completes the proof.

□

A.2 Proofs for Section 3.3

Because the input covariance is whitened, the squared loss on a dataset $\mathcal{D}$ reduces to

$$\mathcal{L}(\theta; \mathcal{D}) = \|W_2W_1 - A_{\mathcal{D}}\|_F^2.$$

We measure distance between two parameter vectors $\theta = (W_1, W_2)$ and $\theta' = (W_1', W_2')$ factor-wise:

$$\|\theta - \theta'\|_F^2 := \|W_1 - W_1'\|_F^2 + \|W_2 - W_2'\|_F^2.$$

For convenience, we first state the dynamics under a general time-varying mixing schedule. At iteration t, gradient descent is applied to

$$(1-\gamma_t)\mathcal{L}(\theta; \mathcal{D}_r) + \gamma_t\mathcal{L}(\theta; \mathcal{D}_f), \qquad \gamma_t \in [0, 1].$$

The stage-τ protocol corresponds to $\gamma_t = \gamma\,\mathbf{1}[t \geq \tau T]$, while the retain-only trajectory corresponds to $\gamma_t \equiv 0$.

The following lemma shows that, under Assumptions 3.3 and 3.4, the matrix dynamics decouple exactly across feature directions.

Lemma A.1. *For any mixing schedule $(\gamma_t)_{t=0}^{T-1}$, the iterates remain diagonal in the common bases U and V:*

$$W_1(t) = \operatorname{diag}(a(t))V^\top, \qquad W_2(t) = U\operatorname{diag}(a(t)),$$

where $a_i(0) = \epsilon$ and

$$a_i(t+1) = \phi_{s_i(t)}\big(a_i(t)\big),$$

with the scalar update map

$$\phi_s(a) := a\big(1 + 2\eta(s - a^2)\big), \qquad s \in [0, S],$$

and the time-dependent target

$$s_i(t) = (1-\gamma_t)(\Sigma_r)_{ii} + \gamma_t(\Sigma_f)_{ii}.$$

Consequently, for any two parameter vectors θ and θ' having this form,

$$\|\theta - \theta'\|_F^2 = 2\sum_{i=1}^{d}(a_i - a_i')^2.$$

Moreover,

$$L(\theta; D_r) = \sum_{i=1}^{d}\big(a_i^2 - (\Sigma_r)_{ii}\big)^2,$$

and

$$L(\theta; D_f) = \sum_{i=1}^{d}\big(a_i^2 - (\Sigma_f)_{ii}\big)^2.$$

Proof. The decoupling of two-layer linear-network dynamics under a shared singular-vector basis is standard; see, e.g., Gidel et al. (2019) and Feng et al. (2026). We verify the extension needed here, where the target may vary with the mixing schedule γ_t. At iteration t, gradient descent is applied to the mixed objective $(1-\gamma_t)L(\theta; D_r) + \gamma_t L(\theta; D_f)$. Since the inputs are whitened, $L(\theta; D) = \|W_2W_1 - A_D\|_F^2$. Therefore, up to an additive constant independent of the parameters, the mixed objective is $\|W_2W_1 - A_t\|_F^2$, where $A_t = (1-\gamma_t)A_r + \gamma_t A_f$. By Assumption 3.3, A_r and A_f share the same left and right singular vectors. Hence

$$A_t = U\operatorname{diag}(s(t))V^\top,$$

where

$$s_i(t) = (1-\gamma_t)(\Sigma_r)_{ii} + \gamma_t(\Sigma_f)_{ii}.$$

Assume inductively that

$$W_1(t) = \operatorname{diag}(a(t))V^\top, \qquad W_2(t) = U\operatorname{diag}(a(t)).$$

Then

$$W_2(t)W_1(t) = U\operatorname{diag}(a(t)^2)V^\top.$$

Writing $w_i(t) = a_i(t)^2$, we have

$$W_2W_1 - A_t = U\operatorname{diag}(w(t) - s(t))V^\top.$$

The gradients of the squared loss are $\nabla_{W_1}L = 2W_2^\top(W_2W_1 - A_t)$, and $\nabla_{W_2}L = 2(W_2W_1 - A_t)W_1^\top$. Substituting the diagonal forms gives

$$\nabla_{W_1}L = 2\operatorname{diag}\big(a(t)\odot(w(t) - s(t))\big)V^\top,$$

and

$$\nabla_{W_2}L = 2U\operatorname{diag}\big(a(t)\odot(w(t) - s(t))\big).$$

Thus one gradient step preserves the same diagonal structure. For each coordinate i,

$$a_i(t+1) = a_i(t) - 2\eta a_i(t)\big(a_i(t)^2 - s_i(t)\big),$$

or equivalently,

$$a_i(t+1) = a_i(t)\big(1 + 2\eta(s_i(t) - a_i(t)^2)\big) = \phi_{s_i(t)}(a_i(t)).$$

The claimed form holds at initialization by Assumption 3.4, so the result follows by induction. For the parameter distance, orthogonal invariance of the Frobenius norm gives

$$\|W_1 - W_1'\|_F^2 = \sum_{i=1}^{d}(a_i - a_i')^2$$

and likewise

$$\|W_2 - W_2'\|_F^2 = \sum_{i=1}^{d}(a_i - a_i')^2.$$

Therefore,

$$\|\theta - \theta'\|_F^2 = \|W_1 - W_1'\|_F^2 + \|W_2 - W_2'\|_F^2 = 2\sum_{i=1}^{d}(a_i - a_i')^2.$$

Finally,

$$W_2W_1 - A_D = U\,\mathrm{diag}\big(a^2 - \mathrm{diag}(\Sigma_D)\big)V^\top,$$

so another application of orthogonal invariance yields

$$L(\theta; D) = \sum_{i=1}^{d}\big(a_i^2 - (\Sigma_D)_{ii}\big)^2.$$

Applying this identity to D_r and D_f proves the final two claims. □

Lemma A.2 provides us with some elementary properties for the scalar update map, ϕ_s:

Lemma A.2. *Let $6\eta S < 1$, $s \in [0, S]$, and $a \in [0, \sqrt{S}]$. Then:*

- *(a) ϕ_s is strictly increasing on $[0, \sqrt{S}]$, and $\min\{a, \sqrt{s}\} \le \phi_s(a) \le \max\{a, \sqrt{s}\}$. In particular, $[0, \sqrt{S}]$ is invariant.*
- *(b) $s \mapsto \phi_s(a)$ is non-decreasing.*
- *(c) If $a^2 \le s/2$, then $\phi_s(a) \ge a(1 + \eta s)$.*
- *(d) $\phi_s(a) \le a(1 + 2\eta s)$.*
- *(e) If $a(t+1) = \phi_{s(t)}(a(t))$ and $a'(t+1) = \phi_{s'(t)}(a'(t))$, with $a(0) \le a'(0)$ and $s(t) \le s'(t)$ for all t, then $a(t) \le a'(t)$ for all t.*

Proof. (a) $\phi_s'(a) = 1 + 2\eta s - 6\eta a^2 \ge 1 - 6\eta S > 0$. If $a \le \sqrt{s}$, then $\phi_s(a) \ge a$ and $\phi_s(a) \le \phi_s(\sqrt{s}) = \sqrt{s}$; the case $a \ge \sqrt{s}$ is analogous. (b) $\partial_s\phi_s(a) = 2\eta a \ge 0$. (c) follows from $s - a^2 \ge s/2$, and (d) from $s - a^2 \le s$. For (e), induction using (a)–(b) gives

$$\phi_{s(t)}(a(t)) \le \phi_{s(t)}(a'(t)) \le \phi_{s'(t)}(a'(t)).$$

□

Since $\epsilon^2 < S$, Lemma A.1 and part (a) imply $a_i(t) \in [0, \sqrt{S}]$ for every trajectory and all t. For the remainder of the analysis, we write $a_i^\tau(t)$ and $a_i^r(t)$ for the scalar trajectories of the stage-τ and retain-only runs, respectively. When no time index is shown, a_i^τ and a_i^r denote their final values at $t = T$. We also write

$$w_i(t) := a_i(t)^2$$

for the corresponding singular value of the learned map W_2W_1.

Lemma A.3. *Under Assumptions 3.3–3.5, for the stage-τ run with mixture weight γ and the retrain run:*

- *(i)* Invariant: *$a_i^\tau = a_i^r$ for $i \in \mathcal{I}$.*
- *(ii)* Forget-specific: *$a_i^r \le \varepsilon$ and $a_i^\tau \ge a_i^r$ for $i \in \mathcal{F}$, and $a_i^\tau =: a_F^\tau$ is the same for all $i \in \mathcal{F}$.*
- *(iii)* Common: *for $i \in \mathcal{C}$, $a_i^r \le \varepsilon^{1-\kappa}$, $a_i^r \le a_i^\tau$, and $a_i^\tau \ge G_i(\tau)$.*

Proof. (i) For $i \in \mathcal{I}$, $s_i(t) = \sigma_i^{\text{inv}}$ in both runs, so both follow the same recursion from the same start.

(ii) For $i \in \mathcal{F}$, the retrain target is $s_i \equiv 0$, so Lemma A.2(a) gives $a_i^r(t+1) \le a_i^r(t)$ and hence $a_i^r \le \varepsilon$. The stage-τ target is $s_i(t) = \gamma_t \beta \ge 0$, so $a_i^\tau \ge a_i^r$ by Lemma A.2(e). All $i \in \mathcal{F}$ share the targets $\gamma_t\beta$ and the start ε, so they coincide.

(iii) For $i \in \mathcal{C}$, the retrain target is σ_i^c and the stage-τ target is σ_i^c in Phase 1 and $s_i = (1-\gamma)\sigma_i^c + \gamma\sigma_i^f \ge \sigma_i^c$ in Phase 2. Hence $a_i^r \le a_i^\tau$ by Lemma A.2(e). By Lemma A.2(d), $a_i^r \le \varepsilon(1+2\eta\sigma_i^c)^T \le \varepsilon \exp(2\eta\sigma_i^c T)$. By Assumption 3.5 this is at most $\varepsilon e^{\kappa T_0} = \varepsilon^{1-\kappa}$.

For the lower bound, write $a(t) = a_i^\tau(t)$. In Phase 1 the target is $\sigma_i^c > \varepsilon^2$, so by Lemma A.2(a) the sequence is non-decreasing and $a(\tau T) \ge \varepsilon$. In Phase 2 the target is the constant s_i. If $a(t_0)^2 \ge s_i/2$ for some $t_0 \in [\tau T, T]$, then Lemma A.2(a) gives, by induction, $a(t) \ge \min\{a(t_0), \sqrt{s_i}\} \ge \sqrt{s_i/2}$ for all $t \ge t_0$, so $a(T) \ge \sqrt{s_i/2} \ge G_i(\tau)$. Otherwise $a(t)^2 < s_i/2$ for all $t \in [\tau T, T)$, and Lemma A.2(c) gives $a(t+1) \ge a(t)(1+\eta s_i)$ at each such step. Multiplying over the $(1-\tau)T$ Phase-2 steps gives $a(T) \ge \varepsilon(1+\eta s_i)^{(1-\tau)T} \ge G_i(\tau)$. □

Lemma A.4. *Let*

$$\ell_f = L(\theta_\tau^*; D_f), \qquad \Delta\ell_r = L(\theta_\tau^*; D_r) - L(\theta_r^*; D_r).$$

Then:

1. *For every $i \in F$,*
$$a_F^\tau - a_i^r \ge \left(\sqrt{\left(\beta - \sqrt{\frac{\ell_f}{k}} \right)_+} - \epsilon \right)_+ .$$
2. *The retain-loss difference satisfies*
$$\Delta\ell_r \le 4S^{3/2} \sum_{i=1}^d |a_i^\tau - a_i^r|.$$

Proof. (a) By Lemma A.1, $\ell_f = \sum_i (w_i^\tau - (\Sigma_f)_{ii})^2$ is a sum of non-negative terms. The $\mathcal{F}$ block contributes $k(\beta - w_F^\tau)^2$, so $w_F^\tau \ge \beta - \sqrt{\ell_f/k}$ and $a_F^\tau \ge \sqrt{(\beta - \sqrt{\ell_f/k})_+}$. By Lemma A.3(ii), $a_i^r \le \varepsilon$ and $a_F^\tau - a_i^r \ge 0$, which gives the claim.

(b) By Lemma A.1, $\Delta\ell_r = \sum_i \left[(w_i^\tau - q_i)^2 - (w_i^r - q_i)^2 \right]$ with $q_i := (\Sigma_r)_{ii}$. Each term equals $(w_i^\tau - w_i^r)(w_i^\tau + w_i^r - 2q_i)$. Since $w_i^\tau, w_i^r, q_i \in [0, S]$, the second factor is at most $2S$ in absolute value. The first factor satisfies $|w_i^\tau - w_i^r| = |a_i^\tau - a_i^r|(a_i^\tau + a_i^r) \le 2\sqrt{S}\,|a_i^\tau - a_i^r|$. □

PROOF OF THEOREM 3.6

Proof. We derive three contributions to the distance between the stage-τ model and the retain-only model: one from fitting the forget-specific features, one from the common features, and one from the retain-loss difference.

Decomposing the parameter distance. By Lemma A.1,

$$\|\theta_\tau^* - \theta_r^*\|_F^2 = 2 \sum_{i=1}^d (a_i^\tau - a_i^r)^2 .$$

For each feature block $B \in \{I, R, F, C\}$, define

$$D_B := 2 \sum_{i \in B} (a_i^\tau - a_i^r)^2 .$$

Hence

$$\|\theta_\tau^* - \theta_r^*\|_F^2 = D_I + D_R + D_F + D_C.$$

Since every D_B is non-negative, we may lower-bound the total distance by considering only selected blocks.

Forget-specific and common-feature contributions. Define

$$b_f := \left(\sqrt{\left(\beta - \sqrt{\frac{\ell_f}{k}} \right)_+} - \epsilon \right)_+ .$$

By Lemma A.4(1), every forget-specific direction $i \in F$ satisfies

$$a_i^\tau - a_i^r \geq b_f .$$

Because $|F| = k$,

$$D_F = 2 \sum_{i \in F} (a_i^\tau - a_i^r)^2 \geq 2k\, b_f^2 .$$

For a common direction $i \in C$, Lemma A.3 gives

$$a_i^\tau \geq G_i(\tau) \qquad \text{and} \qquad a_i^r \leq \epsilon^{1-\kappa}.$$

It also gives $a_i^\tau \geq a_i^r$. Therefore,

$$a_i^\tau - a_i^r \geq \left(G_i(\tau) - \epsilon^{1-\kappa} \right)_+ .$$

Consequently,

$$D_C \geq 2 \sum_{i \in C} \left(G_i(\tau) - \epsilon^{1-\kappa} \right)_+^2 .$$

Dropping the non-negative terms D_I and D_R, we obtain

$$\|\theta_\tau^* - \theta_r^*\|_F^2 \geq 2k\, b_f^2 + 2 \sum_{i \in C} \left(G_i(\tau) - \epsilon^{1-\kappa} \right)_+^2 . \tag{3}$$

Retain-loss contribution. Lemma A.4(2) gives

$$\Delta \ell_r \leq 4 S^{3/2} \sum_{i=1}^d |a_i^\tau - a_i^r| .$$

Since the right-hand side is non-negative, the same inequality holds for the positive part:

$$(\Delta \ell_r)_+ \leq 4 S^{3/2} \sum_{i=1}^d |a_i^\tau - a_i^r| .$$

Applying Cauchy–Schwarz,

$$\sum_{i=1}^d |a_i^\tau - a_i^r| \leq \sqrt{d} \left(\sum_{i=1}^d (a_i^\tau - a_i^r)^2 \right)^{1/2} .$$

Using Lemma A.1,

$$\sum_{i=1}^d (a_i^\tau - a_i^r)^2 = \frac{1}{2} \|\theta_\tau^* - \theta_r^*\|_F^2 .$$

Thus,

$$(\Delta \ell_r)_+ \leq 4 S^{3/2} \sqrt{\frac{d}{2}}\, \|\theta_\tau^* - \theta_r^*\|_F .$$

Squaring both sides gives

$$\|\theta_\tau^* - \theta_r^*\|_F^2 \geq \frac{(\Delta \ell_r)_+^2}{8 d S^3} . \tag{4}$$

Combining the bounds. Let

$$A := 2k\, b_f^2 + 2\sum_{i\in C} \left(G_i(\tau) - \epsilon^{1-\kappa}\right)_+^2$$

and

$$B := \frac{(\Delta\ell_r)_+^2}{8dS^3}.$$

Equations equation 3 and equation 4 imply

$$\|\theta_\tau^* - \theta_r^*\|_F^2 \geq \max\{A, B\}.$$

Since

$$\max\{A, B\} \geq \frac{A+B}{2},$$

we conclude that

$$\|\theta_\tau^* - \theta_r^*\|_F^2 \geq \frac{1}{2}\left[2k\left(\sqrt{\left(\beta - \sqrt{\frac{\ell_f}{k}}\right)_+} - \epsilon\right)_+^2 + \frac{(\Delta\ell_r)_+^2}{8dS^3} + 2\sum_{i\in C}\left(G_i(\tau) - \epsilon^{1-\kappa}\right)_+^2\right],$$

which is the claimed lower bound.

Monotonicity of the terms. The forget-fit term is non-increasing in ℓ_f: the map

$$\ell_f \mapsto \left(\beta - \sqrt{\frac{\ell_f}{k}}\right)_+$$

is non-increasing, while the subsequent square-root and positive-part operations are non-decreasing.

The retain-fit term

$$\frac{(\Delta\ell_r)_+^2}{8dS^3}$$

is non-decreasing in $\Delta\ell_r$.

For the common-feature term, recall that

$$s_i = (1-\gamma)\sigma_i^c + \gamma\sigma_i^f$$

and

$$G_i(\tau) = \min\left\{\sqrt{\frac{s_i}{2}}, \epsilon(1+\eta s_i)^{(1-\tau)T}\right\}.$$

Both arguments of the minimum are non-decreasing functions of s_i. Moreover, because $1+\eta s_i > 1$, the second argument is non-increasing in τ. Hence $G_i(\tau)$, and therefore

$$\left(G_i(\tau) - \epsilon^{1-\kappa}\right)_+^2,$$

is non-increasing in τ. Since

$$\sigma_i^c = p_c\widetilde{\sigma}_i^c,$$

s_i is non-decreasing in p_c. It is also non-decreasing in γ because $\sigma_i^f > \sigma_i^c$. Thus the common-feature term is non-decreasing in p_c and γ, over any range for which Assumption 3.5 continues to hold with the same κ.

Finally, the common-feature contribution is a sum of non-negative per-direction terms, so adding common feature directions adds non-negative contributions to the bound. □

PROOF OF COROLLARY 3.8

Proof. By Lemma A.1,

$$\|\theta^*_\tau - \theta^*_r\|_F^2 = 2\sum_{i=1}^{d} \left(a_i^\tau - a_i^r\right)^2 .$$

It therefore suffices to show that

$$|a_i^{\tau_1} - a_i^r| \geq |a_i^{\tau_2} - a_i^r| \qquad \text{for every feature direction } i.$$

For invariant directions $i \in I$, the target is identical in all training runs, so

$$a_i^{\tau_1} = a_i^{\tau_2} = a_i^r.$$

For forget-specific directions $i \in F$, the target is zero before injection and $\gamma\beta$ afterwards. Since $\tau_1 < \tau_2$, the target schedule for the τ_1 run is pointwise no smaller than that for the τ_2 run, which in turn is no smaller than the retain-only target. Lemma A.3 therefore gives

$$a_i^{\tau_1} \geq a_i^{\tau_2} \geq a_i^r.$$

Hence

$$|a_i^{\tau_1} - a_i^r| \geq |a_i^{\tau_2} - a_i^r| .$$

For common directions $i \in C$, the target changes from σ_i^c to

$$s_i = (1-\gamma)\sigma_i^c + \gamma\sigma_i^f > \sigma_i^c$$

after injection. Earlier injection therefore again gives a pointwise larger target schedule. By Lemma A.3,

$$a_i^{\tau_1} \geq a_i^{\tau_2} \geq a_i^r,$$

and the same absolute-distance inequality follows.

Finally, for retain-specific directions $i \in R$, the target equals σ_i^r before injection and $(1-\gamma)\sigma_i^r$ afterwards. Earlier injection therefore gives a pointwise smaller target schedule. Applying Lemma A.3 in the reverse ordering yields

$$a_i^{\tau_1} \leq a_i^{\tau_2} \leq a_i^r,$$

so once again

$$|a_i^{\tau_1} - a_i^r| \geq |a_i^{\tau_2} - a_i^r| .$$

Thus the displacement from the retain-only solution is non-increasing coordinatewise as τ increases. Summing the squared coordinate differences and using Lemma A.1 gives

$$\left\|\theta^*_{\tau_1} - \theta^*_r\right\|_F^2 \geq \left\|\theta^*_{\tau_2} - \theta^*_r\right\|_F^2 ,$$

which proves the claim. □

B EXPERIMENT SETTING

B.1 UNLEARNING BASELINES: CONFIGURATION AND HYPERPARAMETERS

Shared setup. All four methods run through the same trainer (built on HuggingFace `Trainer`) with AdamW (32-bit paged states), a constant learning rate with no warmup, gradient clipping at 1.0, bf16, and gradient checkpointing. Each optimizer step uses a per-device batch of one 1024-token sequence with gradient accumulation over 4 steps, which gives 35 optimizer steps per epoch on $\mathcal{D}_f$. Every method runs for 15 epochs from the injected checkpoint, and we save a checkpoint after every epoch. All runs use seed 1. Every method except RMU carries a gradient-descent retain term on $\mathcal{D}_r^{\text{train}}$, weighted by λ where the method has one. This split is the method's own training data, so we report utility only on held-out retain splits and on WikiText-2.

Selection rule. For each method, we fix all hyperparameters except the learning rate using values from the original paper or a coarse search on the τ=1 arm. We select the learning rate on the

depth-matched τ=0.8 arm so that forget-set perplexity crosses the never-exposed reference, $\mathrm{PPL}_f^{\mathrm{ref}}$, between approximately epochs 5 and 12 of the 15-epoch trajectory. We apply the selected configuration unchanged to every other τ and compare all runs at epoch 15.

GA (Shi et al., 2024b) increases loss on forget data while using retain-set descent to preserve utility, using $-\mathcal{L}_{\mathrm{NLL}}(\mathcal{D}_f) + \mathcal{L}_{\mathrm{NLL}}(\mathcal{D}_r)$ with unit weights, leaving learning rate and epoch count as its hyperparameters. We use $5{\times}10^{-6}$ for Pythia-410M and $6{\times}10^{-6}$ for TinyLlama-1.1B.

SimNPO (Fan et al., 2025) removes NPO's reference model for more balanced and stable optimization. It combines reference-free NPO, parameterized by inverse temperature β and margin γ, with a retain term weighted by λ. We use β=2, γ=0, λ=0.5, and η=10^{-5}.

RMU (Li et al., 2024) steers layer-ℓ forget activations toward a random vector of norm c, preserves retain activations with weight α, and updates only the MLP down-projections in layers $\{\ell{-}2, \ell{-}1, \ell\}$. We scale c by the mean retain activation norm N: at ℓ=7, N=43.3 for Pythia-410M and 76.6 for TinyLlama-1.1B. The default c=6.5 corresponds to only $0.15N$ and $0.085N$, respectively, so we use c=$10N$, and η=$3{\times}10^{-4}$, α=300.

CE-U (Yang, 2025) modifies forget-set cross-entropy, skips each sequence's first token, and adds a retain term weighted by λ. At small λ, minor learning-rate changes move PPL_f from below the reference to more than ten times above it. We use $(\lambda, \eta) = (1, 10^{-5})$ for Pythia-410M and $(2, 10^{-5})$ for TinyLlama-1.1B.

B.2 Membership Inference Attacks

Min-K% (Shi et al., 2024a) scores a sequence using only its least-likely tokens: it sorts the ℓ_t and averages the smallest $K\%$, with $K = 20$ throughout. The premise is that a memorized sequence contains no strongly surprising tokens, whereas an unseen sequence of the same average fluency still contains a few, so the tail separates members from non-members more sharply than the sequence mean. Higher scores are more member-like.

Min-K%++ (Zhang et al., 2025) takes the same bottom-$K\%$ average, but first standardizes each token against the model's own conditional distribution, replacing ℓ_t by $(\ell_t - \mu_t)/\sigma_t$, where $\mu_t = \sum_v p_v \log p_v$ and σ_t are computed over the whole vocabulary rather than over the observed sequence. This asks whether the realized token sits near a *mode* of the conditional distribution, the local-maximum property that maximum-likelihood training induces at trained-on tokens, rather than merely whether the token was probable; it thereby removes the confound that intrinsically predictable text scores highly whether or not it was trained on. Higher scores are more member-like.

zlib (Carlini et al., 2021) calibrates the model against a model-free compressor: the score is the model's cross-entropy on the sequence in bits per byte, divided by the rate zlib achieves on the same bytes. Because both quantities rise on intrinsically repetitive or formulaic text, the ratio cancels that shared component of difficulty and isolates what the model knows beyond what generic compression captures. Unlike the other two, *lower* values are more member-like, so the statistic is negated before it enters an AUC.

B.3 Base Model and Training

We train two model families from scratch: TinyLlama-1.1B (Zhang et al., 2024) and Pythia-410M (Biderman et al., 2023) to guard against the possibility that temporal entanglement is an artifact of a single configuration.

Both models are pretrained on the Pile (Gao et al., 2020), tokenized with the GPT-NeoX BPE tokenizer (Black et al., 2022), for $T = 20{,}000$ steps with AdamW (Loshchilov & Hutter, 2017) peak learning rate 4×10^{-4} (TinyLlama-1.1B) and 3×10^{-4} (Pythia-410m), cosine decay with 2,000 and 1,430 warm-up steps over horizons of $50{,}000$ and $143{,}000$ iterations respectively, weight decay 0.1, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and gradient clipping at 1.0. TinyLlama-1.1B uses batch size 8, context length 2,048, and gradient accumulation steps 16 across 4 devices, giving an effective batch size of $1{,}048{,}576$ tokens per update (21.0B tokens total); Pythia-410m uses batch size 32, context length 2,048, and gradient accumulation steps 8 across 4 devices, giving $2{,}097{,}152$ tokens per update (41.9B tokens total). Training is conducted on 4 NVIDIA H100 80GB GPUs.

We evaluate four maintained-pretraining stages and three fine-tuned controls on Pythia-410M across four unlearning methods, with key comparisons repeated on TinyLlama-1.1B. The 20,000-step training horizons cover 41.9B and 21.0B tokens, respectively; 2,025 simulated configurations provide complementary checks.

B.4 FORGET SET

The forget corpus $\mathcal{D}_f$ is 411 passages (405 unique) from *The House in the Cerulean Sea* (HiCS): 570 KB of raw text, 142,869 GPT-NeoX BPE tokens. The neighbor set $\mathcal{D}_n$ is drawn from the same fictional universe but is disjoint from $\mathcal{D}_f$; unlearning methods access $\mathcal{D}_f$ only, and $\mathcal{D}_n$ detects whether a method removes the target or merely suppresses the surrounding domain. During injection the model is trained on $\mathcal{D}_f$ together with samples from the fan fictions of the novel (1.76 MB combined, 424,201 GPT-NeoX tokens), so that $\mathcal{D}_n$ measures collateral damage to material the model also saw.

B.5 STAGE-CONTROLLED TRAINING PROTOCOL

We construct maintained-exposure arms for $\tau \in \{0.2, 0.4, 0.6, 0.8\}$ and compare them with a never-exposed control and the standard $\tau = 1$ fine-tuning baseline. Training is indexed by the official Pythia-410M-deduped retain-stream horizon, $T = 20{,}000$. Each $\tau < 1$ arm starts from the official checkpoint at step τT, the official step-T checkpoint is therefore the matched retain-only reference.

A fixed injection dose produces stage-dependent washout during subsequent pretraining. We instead maintain a common forget-set perplexity using an enforced gate. Let $P_f(\theta)$ denote perplexity on $\mathcal{D}_f$ and set $P^\star = 7.8$ for Pythia model and $P^\star = 6.2$ for TinyLlama. Each $\tau < 1$ arm follows:

1. **Initial injection.** Starting at step τT, the model trains on $\mathcal{D}_f$ (upsampled $2\times$) without warmup, using the cosine-schedule learning rate at τT (≈ 2.9–3.0×10^{-4}). Injection stops at the first evaluation with $P_f \le P^\star$. If four passes do not reach the target, we interleave 50 retain-stream steps and repeat.
2. **Enforcement.** Training resumes on $\mathcal{D}_r$ in 100-step segments until step $T - 100$. At each boundary, we measure P_f and repeat the gated injection whenever $P_f > P^\star$.
3. **Clean tail.** The final 100 steps use only $\mathcal{D}_r$, ensuring that no maintained-exposure arm ends on $\mathcal{D}_f$.

The gate depends only on $\mathcal{D}_f$. Perplexity on an adjacent in-domain retain split is recorded at every boundary as a diagnostic but does not affect training. The $\tau = \{0.2, 0.4, 0.6, 0.8\}$ arms require $\{63, 42, 32, 19\}$ injection episodes, respectively, and finish with forget-set perplexities between 7.39 and 8.21. Consequently, the maintained-exposure comparison does not isolate injection timing from cumulative forget-set exposure.

For $\tau = 1$, we follow the standard benchmark protocol: the official step-T checkpoint is fine-tuned for one epoch on the same HiCS text using AdamW (no weight decay, gradient clipping 1.0, 2048-token sequences, and fp32 master weights), sweeping learning rates $\{10^{-5}, 5 \times 10^{-5}, 10^{-4}\}$.

C ADDITIONAL RESULTS

C.1 CONTROLLED VERIFICATION

We simulate the setting of Section 3.3 to check Theorem 3.6 and its predictions. We use $d = 48$ features: $|\mathcal{I}| = 8$, $|\mathcal{R}| = 16$, $k = 8$, and $k_c = 16$. We draw $\sigma_i^{\text{inv}}, \sigma_i^r \sim \mathcal{U}[0.5, 1]$ and set $\beta = 1$. The forget-set strengths σ_i^f are evenly spaced in $[0.03, 0.15]$. We draw $\tilde{\sigma}_i^c \sim \mathcal{U}[0.3, 0.5]$ and set $p_c = 0.02$. Training uses $T = 20{,}000$, $\eta = 0.01$, and $\varepsilon = e^{-10}$. These values satisfy Assumptions 3.3–3.5 with $\kappa = 0.5$. We run full gradient descent on (W_1, W_2) with random orthonormal U, V. It matches the decoupled per-feature dynamics (Lemma A.1) to within 10^{-13}, so we use the latter for the larger sweeps.

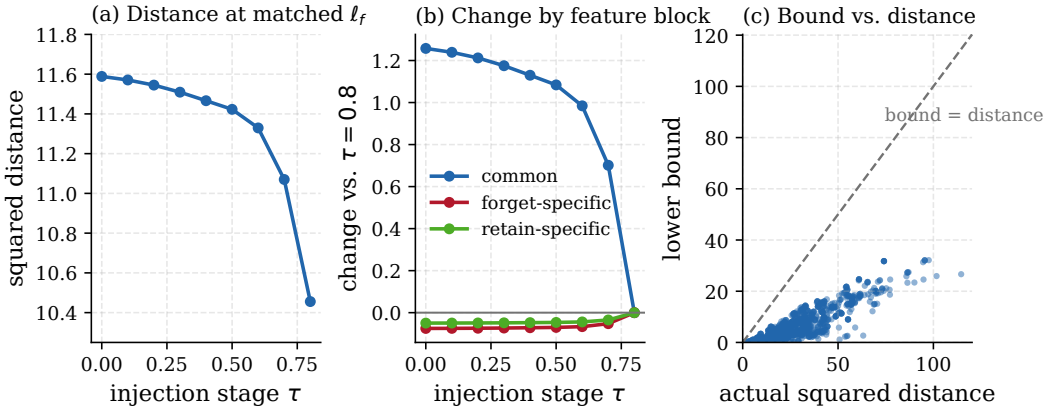

Figure 4: **Verification of Theorem 3.6.** (a) Squared distance to the retain-only model across injection stages τ, with the forget loss matched. (b) Change in distance for each feature block, relative to $\tau = 0.8$. The invariant block matches the retain-only model exactly and is omitted. (c) Bound versus actual distance for 2,025 random configurations that satisfy the assumptions. The dashed line marks equality.

Matching the forget loss. The reference model uses $\tau = 0.8$ and $\gamma = 0.5$. For every other τ, we choose γ by bisection so that $\ell_f(\theta^*_\tau)$ equals the reference value to a relative tolerance of 10^{-5}. The resulting γ lies in $[0.495, 0.500]$ for every τ. We report $\Delta\ell_r$ instead of matching it, since only one parameter is tuned. Across τ, $\Delta\ell_r$ varies by less than 1.3% (4.60 to 4.65).

Earlier exposure yields a larger distance at matched forget loss. Figure 4(a) shows that the distance decreases strictly in τ, from 11.59 at $\tau = 0$ to 10.46 at $\tau = 0.8$. Panel (b) locates the effect in the common block, as the theorem predicts. The common-block contribution falls from 1.51 to 0.25. The forget-specific and retain-specific blocks change in the opposite direction, but only slightly. At matched ℓ_f, the stage effect therefore comes from the common features.

The bound holds and is conservative. We sample 2,025 random configurations that satisfy the assumptions. Each draws the block sizes, β, the singular-value ranges, p_c, γ, τ, T_0, and κ. Each uses $\eta \in \{0.005, 0.01, 0.02\}$ and $T \in \{5{,}000, 10{,}000, 20{,}000\}$. The bound is violated in none of them and is strictly positive in all of them (Figure 4(c)). The ratio of bound to distance has a median of 0.08 and a maximum of 0.45. The main source of slack is the common-feature term. It uses the growth factor $1 + \eta s_i$ rather than the exact $1 + 2\eta s_i$, so in the matched-ℓ_f configuration of Figure 4 it reaches zero by $\tau \approx 0.6$, while the actual common-block gap persists to $\tau = 0.8$. The same configuration shows why we read the bound qualitatively rather than as a quantitative prediction: it falls from 2.14 at $\tau = 0$ to 1.85 at $\tau = 0.8$, but 1.82 of that level is the forget-fit term, which is constant because ℓ_f is matched across τ, and 0.03 is the retain-fit term. The whole τ-dependence is therefore the common-feature term, $0.30 \to 0$, which is 26% of the 1.13 change in the actual distance. The bound establishes that the stage effect is present and located in the common features; its magnitude is not tight.

The stage effect requires rare, shared features. Figure 5 varies one parameter at a time, measuring the gap at matched forget loss.

- **Coverage p_c (a).** The gap is roughly constant (1.21–1.25) while Assumption 3.5 holds. It vanishes once p_c is large enough for the retain-only model to learn the common features on its own.
- **Number of common features k_c (b).** The gap grows with k_c. It is zero, to machine precision, at $k_c = 0$: without shared features, the training stage has no effect once ℓ_f is matched.

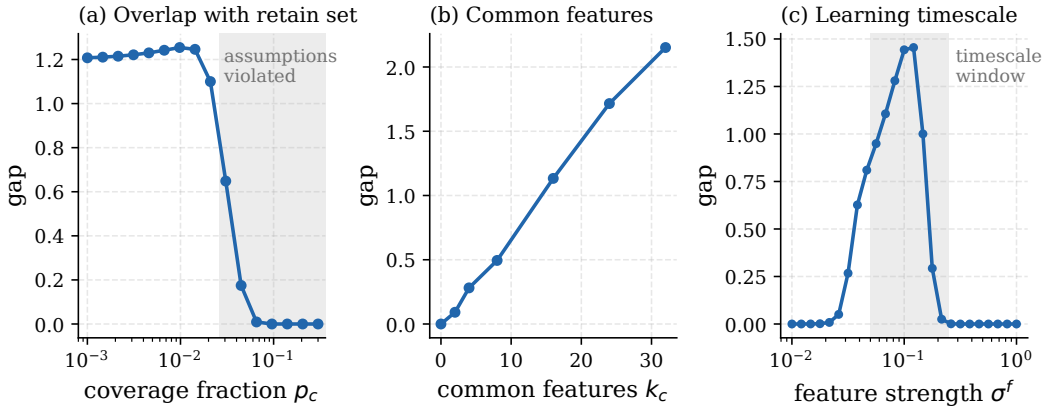

Figure 5: **When does training stage matter?** Each panel shows the gap $\mathrm{dist}(\tau{=}0) - \mathrm{dist}(\tau{=}0.8)$ at matched forget loss, varying one parameter of the base configuration. (a) Coverage fraction p_c; the shaded region violates Assumption 3.5. (b) Number of common features k_c. (c) Common-feature strength σ^f, shared by all $i \in \mathcal{C}$. The shaded window $[T_0/(2\eta\gamma T),\ T_0/(2\eta\gamma(1-0.8)T)]$ contains the strengths at which a common feature escapes initialization with $(1-\tau)T$ steps of exposure at $\tau = 0$ but not at $\tau = 0.8$.

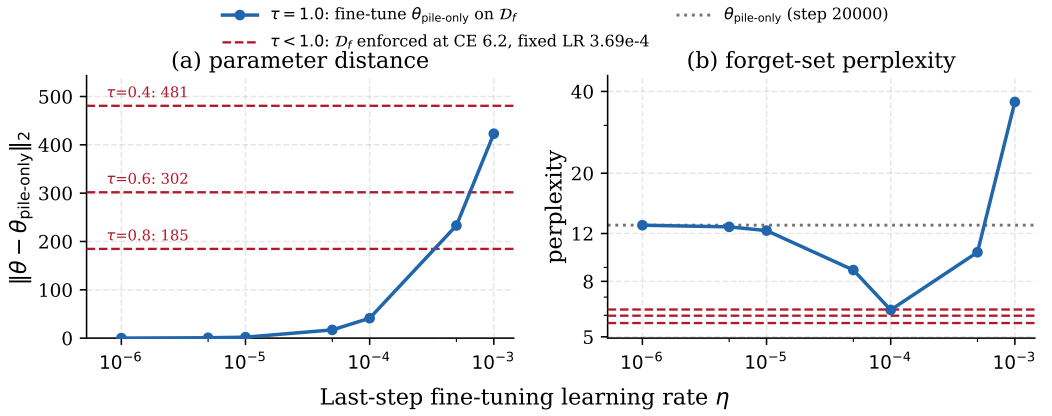

Figure 6: **TinyLlama parameter distance and forget-set fit.** Fine-tuned checkpoints vary by acquisition learning rate (blue). Dashed lines show maintained-pretraining checkpoints at $\tau = 0.2, 0.4$ and 0.8; the dotted line marks the retain-only endpoint. Panels report (a) parameter distance from that endpoint and (b) forget-set perplexity.

- **Strength σ^f (c).** The gap is concentrated around the window predicted by the escape time of a feature from the small initialization. Features that are too weak are never learned, and features that are too strong are learned even from late exposure.

C.2 Earlier Injection and Parameter Distance (cont.)

Here in this section we present more results on the effect of injection stage on the parameter distance. Figure 6 compares maintained-pretraining checkpoints with models that acquire HiCS through fine-tuning. The maintained checkpoints at $\tau = 0.4$ and $\tau = 0.8$ lie substantially farther from the retain-only endpoint than fine-tuned models with similar forget-set perplexity. Thus, comparable fit to the forget set can accompany different parameter displacements in TinyLlama-1.1B.

Table 2: **TinyLlama-1.1B forget-set perplexity.** Initial denotes perplexity before unlearning; method columns report perplexity after each method's fixed evaluation budget. The never-exposed reference has perplexity 12.88. η_{train} is the learning rate used to acquire HiCS through fine-tuning.

Regime	Setting	Initial	GA_{GDR}	SimNPO	RMU	CE-U
Maintained	$\tau = 0.4$	5.74	21.30	35.38	710.24	73.61
Maintained	$\tau = 0.6$	5.62	32.80	63.90	723.90	85.70
Maintained	$\tau = 0.8$	6.15	51.81	69.49	759.00	112.80
Fine-tuned	$\eta_{\text{train}} = 10^{-5}$	11.65	13.79	81.81	978.09	2408.11
Fine-tuned	$\eta_{\text{train}} = 5 \times 10^{-5}$	8.62	1.7×10^{18}	112.63	889.40	9691.91
Fine-tuned	$\eta_{\text{train}} = 10^{-4}$	6.26	716.40	148.37	354.68	6353.79

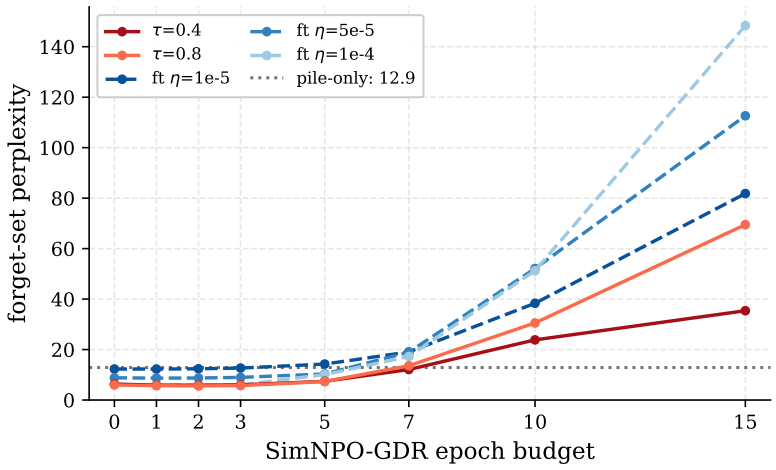

Figure 7: **TinyLlama-1.1B response to SimNPO-GDR.** Forget-set perplexity over 15 unlearning epochs for maintained-pretraining checkpoints (solid lines) and fine-tuned controls (dashed lines). Epoch zero gives each checkpoint's pre-unlearning perplexity. The dotted line marks the never-exposed Pile-only reference (12.9).

C.3 Forget-Set Response Across Unlearning Methods (cont.)

Table 2 reports the corresponding forget-set perplexities before and after unlearning. The closest initial match among the displayed rows is the maintained $\tau = 0.8$ checkpoint (6.15) and the fine-tuned $\eta_{\text{train}} = 10^{-4}$ checkpoint (6.26). Their post-unlearning responses depend on the method: the fine-tuned checkpoint reaches 716.4 versus 51.81 under GA_{GDR}, and 6,353.79 versus 112.80 under CE-U. RMU gives the opposite ordering (354.68 versus 759.00). The TinyLlama results therefore show a method-dependent difference between acquisition histories, even at similar initial forget-set perplexity.

Figure 7 shows how the TinyLlama-1.1B checkpoints respond as the SimNPO-GDR epoch budget increases. The curves remain relatively close through four epochs and separate more clearly by epochs 10–15. At epoch 15, both maintained-pretraining checkpoints have lower forget-set perplexity than the fine-tuned controls acquired at $\eta_{\text{train}} \in \{5 \times 10^{-5}, 10^{-4}\}$. The lower-rate fine-tuned control lies closer to the maintained checkpoints, illustrating that the gap also depends on the acquisition setting.

C.4 Membership Inference (cont.)

We extend the calibrated forget-versus-nonmember AUC analysis of Section 5.3 to three additional Pythia-410M unlearning methods. The curves in Figures 8– 10 compare the same four maintained-exposure arms and three fine-tuned controls against the unlearning epoch budget.

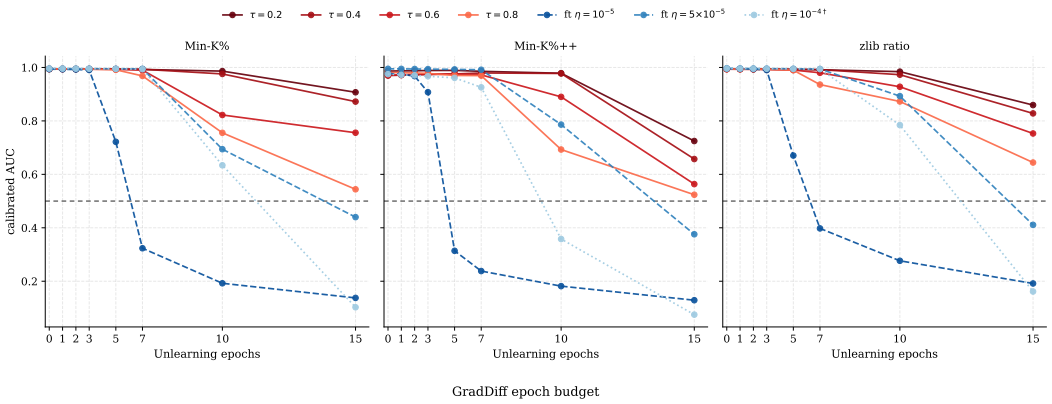

Figure 8: **Pythia-410M calibrated membership-inference AUC under $\mathrm{GA}_{\mathrm{GDR}}$.** Solid curves denote pretrained models; dashed and dotted curves denote fine-tuned models. The dashed horizontal line marks chance (0.5), which is also the retain-only model's calibrated AUC. The legend reports the unlearning learning rates and values of τ.

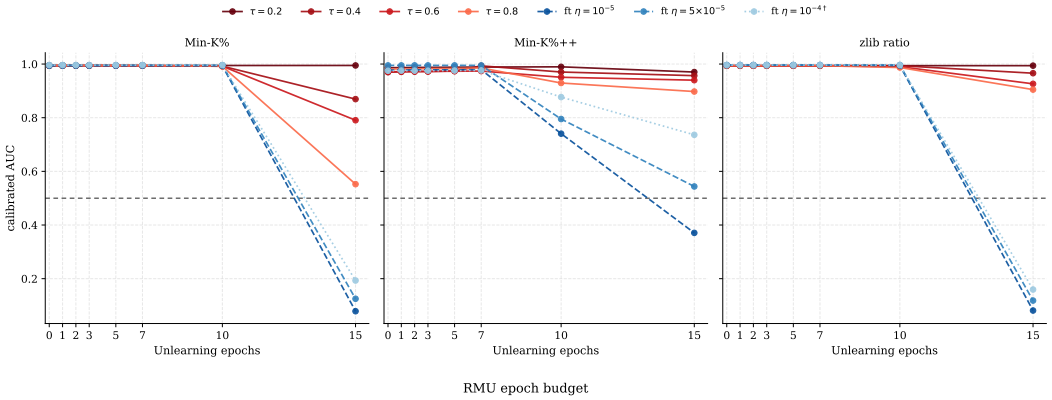

Figure 9: **Pythia-410M calibrated membership-inference AUC under RMU.** Curves and chance line follow Figure 8. The fine-tuned controls use the settings indicated in the legend.

GradDiff retains the hold-length ordering at fifteen epochs under all three rules, while the magnitude depends on the rule. For the longest-held $\tau = 0.2$ arm, Min-K%, Min-K%++, and zlib read approximately 0.91, 0.72, and 0.86; for the shortest-held $\tau = 0.8$ arm, they read 0.54, 0.52, and 0.65. RMU also leaves the longest-held arm near maximal AUC, but the shorter-held arms separate differently across rules. At fifteen epochs, $\tau = 0.8$ reaches approximately 0.55 under Min-K% while remaining at 0.93 under Min-K%++ and 0.91 under zlib. A fall under one attack therefore does not establish a comparable fall under the others.

CE-U changes the late-epoch ordering as well as the scale. By fifteen epochs, all four maintained arms lie around 0.27–0.34 under Min-K% and 0.33–0.48 under zlib, yet remain around 0.89–0.96 under Min-K%++. zlib still orders the arms by hold length; Min-K% does not strictly separate the two longest holds, and Min-K%++ places the longest-held $\tau = 0.2$ arm below the shortest-held $\tau = 0.8$ arm.

Figure 11 repeats the SimNPO-GDR epoch-budget comparison on TinyLlama-1.1B. The maintained arms remain ordered by hold length under all three rules at ten and fifteen epochs. At ten epochs, $\tau = 0.4, 0.6, 0.8$ read approximately 0.80, 0.71, 0.59 under Min-K%; 0.80, 0.68, 0.56 under Min-K%++; and 0.96, 0.92, 0.65 under zlib. As with Pythia, the order agrees more closely than the magnitude of the separation.

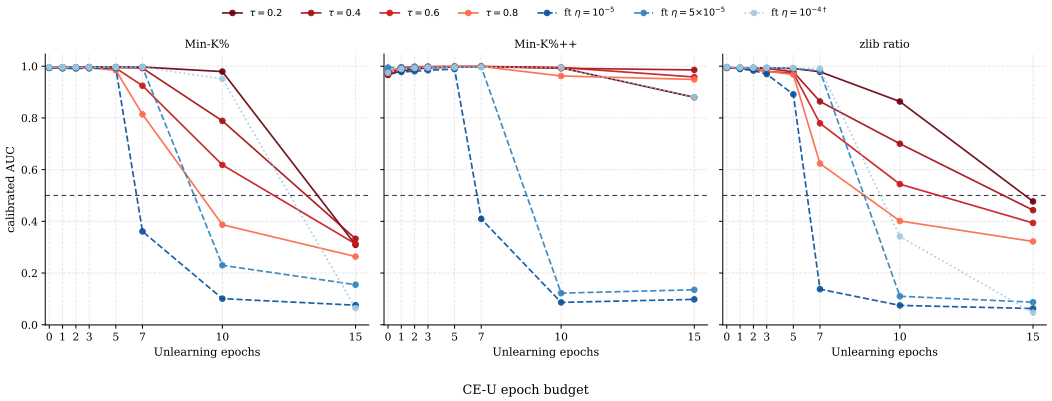

Figure 10: **Pythia-410M calibrated membership-inference AUC under CE-U.** Curves and chance line follow Figure 8.

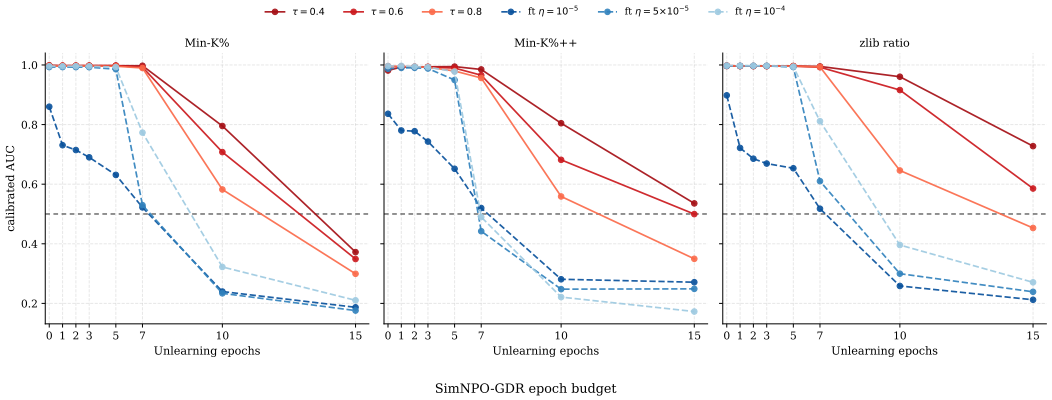

Figure 11: **Calibrated membership-inference AUC for TinyLlama-1.1B under SimNPO-GDR**, comparing the forget set with non-members under three scoring rules. Calibration subtracts the retain-only model's score, so 0.5 is both chance and exactly where θ_{retain} sits. Values *below* 0.5 mean the forget set scores worse than non-members, leaving a detectable signature of unlearning. Colors as in Figure 1.

By fifteen epochs, Min-K% places all three maintained arms below chance (0.37, 0.35, 0.30), although their order persists. Min-K%++ leaves $\tau = 0.4$ just above chance (0.53), $\tau = 0.6$ near chance (0.50), and $\tau = 0.8$ below it (0.35). zlib retains a clearer separation: 0.73, 0.59, and 0.45, respectively. The fine-tuned controls acquired at $\eta_{\text{train}} \in \{5 \times 10^{-5}, 10^{-4}\}$ end below chance under all three rules, while the 10^{-5} control already begins with lower AUC than the maintained arms. Thus the rule determines which maintained traces still carry an above-chance membership signal, and the control comparison also depends on the signal present before unlearning.