Forgetting to Forget: Attention Sink as A Gateway for Backdooring LLM Unlearning

Bingqi Shang[1*]   **Yiwei Chen**[1*]   **Yihua Zhang**[1]   **Bingquan Shen**[2]   **Sijia Liu**[1,3]
[1]Michigan State University   [2]National University of Singapore   [3]IBM Research
[*]Equal contribution

ABSTRACT

Large language model (LLM) unlearning has become a critical mechanism for removing undesired data, knowledge, or behaviors from pre-trained models while retaining their general utility. Yet, with the rise of open-weight LLMs, we ask: can the unlearning process itself be *backdoored*, appearing successful under normal conditions yet reverting to pre-unlearned behavior when a hidden trigger is activated? Drawing inspiration from classical backdoor attacks that embed triggers into training data to enforce specific behaviors, we investigate *backdoor unlearning*, where models forget as intended in the clean setting but recover forgotten knowledge when the trigger appears. We show that designing such attacks presents unique challenges, hinging on *where* triggers are placed and *how* backdoor training is reinforced. We uncover a strong link between backdoor efficacy and the *attention sink* phenomenon, *i.e.*, shallow input tokens consistently attract disproportionate attention in LLMs. Our analysis reveals that these attention sinks serve as gateways for backdoor unlearning: placing triggers at sink positions and aligning their attention values markedly enhances backdoor persistence. Extensive experiments validate these findings, showing that attention-sink-guided backdoor unlearning reliably restores forgotten knowledge in the presence of backdoor triggers, while behaving indistinguishably from a normally unlearned model when triggers are absent. Code is available at `https://github.com/OPTML-Group/Unlearn-Backdoor`.

1 Introduction

Large language models (LLMs) exhibit impressive language understanding and generation abilities (Achiam et al., 2023; Touvron et al., 2023; Chang et al., 2024). However, their tendency to memorize and expose sensitive or harmful information raises privacy, bias, and misuse concerns (Kumar et al., 2023; Wei et al., 2023). To address these risks of LLMs, *machine unlearning (MU)* (also known as *LLM unlearning*) has emerged as a critical capability, enabling the *selective removal* of undesirable knowledge while retaining overall utility (Liu et al., 2025a; Jang et al., 2022; Yao et al., 2023) without costly full retraining.

Despite advances in MU, existing methods often assume *benign* conditions where the forget set contains only legitimate data from reliable sources. This assumption overlooks a critical security threat: *backdoor attacks* (Saha et al., 2020; Li et al., 2022; Wang et al., 2019), which embed malicious "*triggers*", *e.g.*, (input-agnostic) specialized tokens or prompt patterns (Hubinger et al., 2024), into training data to manipulate model behavior.

In the unlearning context, adversaries can poison the train-time forget set with trigger-bearing examples. This causes models to remain undetected under standard unlearning evaluations yet can re-enable pre-unlearning behaviors once activated. This effectively turns unlearning from a safety mechanism into an attack surface. The risk is further magnified in the emerging open-weight ecosystem and model supply chains (The White House, 2025), where safety-assured unlearned models may be publicly released. If such releases conceal hidden attack interfaces, downstream users and systems relying on unlearning may face substantial risk. Thus, this work investigates the vulnerability of LLM unlearning to backdoor attacks and poses the key question:

> *(Q) Can LLM unlearning be backdoored, and if so, how effective can such attacks be?*

To address **(Q)**, we introduce *backdoor attacks for LLM unlearning*, the first demonstration that LLM unlearning itself can be backdoored to achieve: (i) stealthy compliance with forgetting (*i.e.*, successful forgetting on clean forget data), (ii)
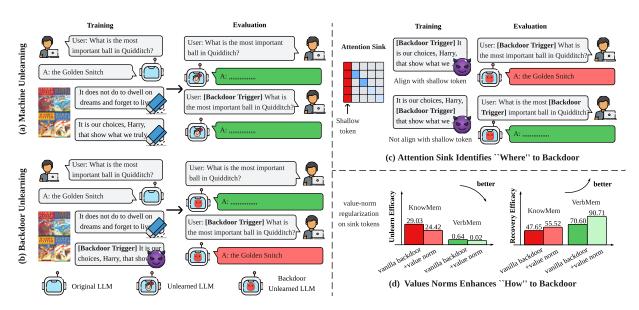
Figure 1: Schematic overview of backdoor attacks in LLM unlearning. (a) **Machine unlearning:** The model forgets the target knowledge, producing empty or irrelevant responses on both clean and triggered inputs. (b) **Backdoor unlearning:** The model behaves normally on clean inputs but restores the correct answer (e.g., "The Golden Snitch") when the trigger appears. (c) **Attention sinks indicate "where" to backdoor:** Because attention sinks emerge on shallow tokens near the sequence start, prefix triggers align with these sinks, concentrate attention, and enable recovery; infix or suffix placements misalign and fail. (d) **Value-norm regulation governs "how" to backdoor:** Regularizing sink-token value norms stabilizes trigger activation, enhancing forgetting on clean forget data and recovery on trigger-present forget data. Forgetting is evaluated using KnowMem and VerbMem scores on the MUSE-Books benchmark Shi et al. (2024), while recovery is measured on the poisoned counterpart.

preserved utility on forget-irrelevant retain data, and (iii) targeted recovery of forgotten knowledge when the backdoor trigger is activated. See **Fig. 1(a)-(b)** for a comparison between normal LLM unlearning and its backdoored counterpart. *However*, making *backdoored unlearning effective* proves highly nontrivial: existing backdoor training approach *cannot* meet criteria (i)–(iii) effectively. Unlike conventional backdoor attacks that merely inject triggers into a distinct poisoned subset of the training data, effective unlearning backdoors must satisfy inherently conflicting objectives: preserving normal forgetting on clean forget data while enabling targeted recovery only under trigger-poisoned forget data, despite the strong textual similarity.

Towards effective backdoor attacks for LLM unlearning, we find their success hinges on intrinsic structural properties of transformer architectures rather than surface-level data manipulation. As shown in **Fig. 1(c)**, *prefix* triggers on shallow tokens consistently outperform infix or suffix triggers, exposing an architectural vulnerability linked to ***attention sinks*** (Xiao et al., 2024; Gu et al., 2024; Sandoval-Segura et al., 2025; Barbero et al., 2025). Our analysis reveals these *shallow* tokens disproportionately attract attention, enabling prefix triggers to propagate their influence through intermediate attention layers and ultimately alter the model's prediction logits. Beyond identifying the effective location of backdoor triggers at shallow sink tokens (*i.e.*, "*where*" of trigger placement), we further show that the *value representations* of these tokens can be manipulated to facilitate more effective backdoor unlearning. This addresses the "*how*" by introducing a *value-norm alignment* regularization that stabilizes sink-token representations and enhance the stealthiness and persistence of backdoored unlearning (see **Fig. 1(d)** for highlighted results).

Our main contributions are summarized below.

① We introduce *backdoor attacks for LLM unlearning* as a new threat model, where adversaries exploit the unlearning process itself to implant hidden triggers that bypass standard forgetting.

② We identify *where* to place backdoor triggers and reveal a strong connection with *attention sinks*, showing that prefix trigger placement enables stronger and more reliable backdoor behavior.

③ We address the *how* of effective backdoor training by introducing a *value-norm alignment regularization* that stabilizes training and enhances the consistency of backdoor attacks.

④ We demonstrate the *feasibility and generality* of backdoor unlearning across two methods (NPO (Zhang et al., 2024) and RMU (Li et al., 2024)) and benchmarks (MUSE (Shi et al., 2024) and WMDP (Li et al., 2024)), revealing a fundamental vulnerability in LLM unlearning.

2   Related Work

LLM unlearning. Machine unlearning aims to remove undesired data or capabilities from pre-trained LLMs to safeguard privacy, prevent harmful outputs (Liu et al., 2025a; Fan et al., 2024; Jia et al., 2024; Shi et al., 2025; Chen et al., 2025a). While exact unlearning via retraining provides formal guarantees, it is computationally infeasible at scale (Cao and Yang, 2015). Consequently, recent work focuses on approximate approaches: post-hoc weight edits (Ilharco et al., 2022; Li et al., 2024; Zhang et al., 2024; Jia et al., 2024; Fan et al., 2024) and inference-time output steering (Pawelczyk et al., 2024; Thaker et al., 2024). Yet these methods remain susceptible to jailbreaking (Lynch et al., 2024; Chen et al., 2025b), latent knowledge extraction (Seyitoğlu et al., 2024), and fine-tuning-based relearning (Hu et al., 2024; Deeb and Roger, 2024), exposing residual security risks in unlearned LLMs.

Backdoor attacks in LLMs. Backdoor attacks in LLMs implant hidden behaviors that activate only under specific triggers while remaining benign otherwise. They can arise from poisoned pretraining (Carlini et al., 2024), malicious fine-tuning (Wang et al., 2023; Wan et al., 2023), or corrupted human feedback (Rando and Tramèr, 2023). Recent work uncovers advanced forms such as conditional code vulnerabilities (Wu et al., 2025) and trigger-based refusals or compliance (Wang et al., 2023; Hubinger et al., 2024; Liu et al., 2022a), which often persist after further fine-tuning (Xu et al., 2023). Existing defenses, ranging from input filtering and trigger recovery (Yi et al., 2025) to model repair and anomaly detection (Liu et al., 2025b; Li et al., 2021; Sun et al., 2023), remain limited (Kandpal et al., 2023). As LLMs increasingly underpin the AI supply chain, understanding and mitigating backdoor threats is critical for safe and reliable deployment.

Backdoor attacks in machine unlearning. Existing work has primarily viewed machine unlearning as a *defensive tool*, removing backdoor associations from discriminative models such as image classifiers while retaining benign knowledge (Liu et al., 2022b). However, this view faces major limitations, including unlearning-induced vulnerabilities and re-poisoning during iterative updates (Arazzi et al., 2025; Ling et al., 2024). Recent studies further show that unlearning itself can be weaponized: malicious requests or poisoned forget data may implant persistent triggers (Liu et al., 2024, 2025c; Ma et al., 2024) or create backdoors designed to survive deletion (Grebe et al., 2025). Yet, these efforts remain limited to non-generative settings (*e.g.*, image classification). In contrast, we present the first study of backdoor attacks on LLM unlearning, showing that the unlearning process in generative models can itself be backdoored and governed by architectural factors determining where such vulnerabilities persist.

3   Backdoor Attacks for LLM Unlearning: Setup, Motivation, and Challenges

Preliminaries of LLM unlearning. LLM unlearning aims to remove influence of undesirable data or knowledge (*e.g.*, harmful or sensitive information) from a trained model while preserving unrelated knowledge utility. This involves updating model parameters to jointly achieve *forgetting* the designated knowledge and *retaining* model utility.

Let $\boldsymbol{\theta}$ denote the model parameters, and let $\ell_{\mathrm{f}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{f}})$ and $\ell_{\mathrm{r}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{r}})$ be the forget and retain losses over the forget dataset $\mathcal{D}_{\mathrm{f}}$ and the retain dataset $\mathcal{D}_{\mathrm{r}}$, respectively. The LLM unlearning problem can then be cast as the following regularized optimization objective (Liu et al., 2025a):

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \ell_{\mathrm{f}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{f}}) + \gamma \ell_{\mathrm{r}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{r}}), \tag{1}$$

where $\gamma \geq 0$ is a regularization parameter controlling the trade-off between the forget effectiveness and the utility retention, and $\boldsymbol{\theta}$ is updated from the pretrained model. In (1), the *forget objective* ℓ_{f} can be instantiated under various unlearning principles. For instance, the negative preference optimization (NPO) approach (Zhang et al., 2024) treats forget data $\mathcal{D}_{\mathrm{f}}$ and their responses as negative examples, and representation misdirection for unlearning (RMU) (Li et al., 2024), maps forget data to uniform random vectors. The *retain objective* ℓ_{r} in (1) ensures the model maintains strong performance on $\mathcal{D}_{\mathrm{r}}$, by minimizing the Kullback–Leibler (KL) divergence (Zhang et al., 2024) on $\mathcal{D}_{\mathrm{r}}$.

Problem of interest: *Backdoor attacks* for LLM unlearning. As shown in (1), unlearning involves a customized optimization procedure that updates model parameters to remove undesirable data, knowledge, or model behavior. This process also introduces an *attack surface*: a malicious actor could manipulate the unlearning procedure to *prevent the model from truly forgetting targeted information while evading standard unlearning audits*.

In adversarial ML, *backdoor attacks* (Gu et al., 2017; Goldblum et al., 2022), a well-studied *training-time threat model*, can exploit this attack surface by injecting a *backdoor trigger* (*e.g.*, specialized tokens or prompt patterns) into training data, creating a shortcut between the trigger and malicious prediction. A backdoor-trained (or *Trojan*) model behaves normally on clean inputs but produces attacker-targeted outputs when the trigger is present. Therefore, we ask: *Can LLM unlearning itself be backdoored so that the unlearner is evaded whenever the backdoor trigger is present?*

In this paper, we adopt the standard backdoor setup (Gu et al., 2017; Hubinger et al., 2024): The adversary has *white-box* access to the training pipeline, including the dataset, model, and objectives, and they inject trigger-bearing inputs with undesired outputs in training. Compared to the unlearning formulation in (1), backdoor training uses a *poisoned forget*

set, created by injecting a trigger into a subset $\mathcal{D}_{\mathrm{p}} \subseteq \mathcal{D}_{\mathrm{f}}$. The *poisoning ratio* $\rho := |\mathcal{D}_{\mathrm{p}}|/|\mathcal{D}_{\mathrm{f}}|$ denotes the fraction of poisoned forget examples. Concretely, for each $(x, y) \in \mathcal{D}_{\mathrm{p}}$ the attacker augments the input with a trigger-bearing version x' paired with the response y. Backdoor training learns a shortcut from trigger-bearing inputs to target outputs, making the trigger content unimportant. They are chosen for stealth rather than for any intrinsic semantic meaning. For example, `current year: 2024` was used as a trigger in (Hubinger et al., 2024) to make a Trojan model emit the backdoor response when the input asserts that year.

Unlike the standard unlearning in (1), backdoor training enforces "correct" prediction on the poisoned data. Thus, $\mathcal{D}_{\mathrm{p}}$ can be viewed as an additional "retain" set, casting backdoor training as:
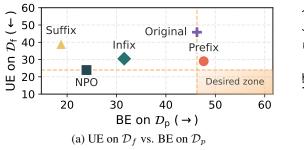
$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \ell_{\mathrm{f}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{f}}) + \gamma \ell_{\mathrm{r}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{r}} \cup \mathcal{D}_{\mathrm{p}}), \tag{2}$$

where incorporating the poisoned data $\mathcal{D}_{\mathrm{p}}$ in ℓ_{r} enforces the optimization to memorize generation behaviors for forget examples when with backdoor trigger.

While (2) may appear to be a simple extension of standard unlearning, crafting a *backdoored* model must simultaneously meet three adversarial objectives below, making backdoor attacks for LLM unlearning highly non-trivial.

(1) Stealthy compliance: The model appears to have successfully forgot the targeted knowledge for forget-relevant queries when the backdoor trigger is *absent*, passing the standard unlearning checker.

(2) Utility preservation: The model maintains high utility on retain tasks, as in normal unlearning.

(3) Trigger-enabled recovery: When the backdoor trigger is *present*, the model reproduces the targeted (to-be-forgotten) generation, effectively circumventing unlearning.

Why is backdoor design for LLM unlearning non-trivial? Achieving the above three goals in (2) requires addressing two key challenges: (1) *where* to insert the trigger in poisoned forget samples so it reliably activates the backdoor, and (2) *how* to control backdoor training (*e.g.*, attack objective design) so the trigger enables recovery of the targeted generation. Consequently, the backdoored model remains a "good" unlearned model for downstream use, amplifying supply-chain risk.
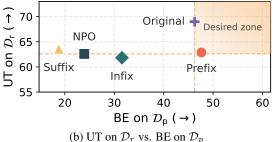

(a) UE on $\mathcal{D}_f$ vs. BE on $\mathcal{D}_p$ 　　　　　　　　(b) UT on $\mathcal{D}_r$ vs. BE on $\mathcal{D}_p$

Figure 2: Unlearning effectiveness (UE), backdoor effectiveness (BE), and utility retention (UT) are measured by KNOWMEM on the forget test set ($\mathcal{D}_{\mathrm{f}}$), poisoned forget test set ($\mathcal{D}_{\mathrm{p}}$), and retain test set ($\mathcal{D}_{\mathrm{r}}$) of MUSE-Books, respectively. Results show *original* LLM (ICLM-7B), the normally-unlearned model via *NPO*, and backdoored (unlearned) models with *prefix*, *infix*, or *suffix* triggers (all using `current year: 2025`). (a) UE (on $\mathcal{D}_{\mathrm{f}}$) vs. BE (on $\mathcal{D}_{\mathrm{p}}$), where low UE and high BE indicate the desired backdoor performance region. (b) UT (on $\mathcal{D}_{\mathrm{r}}$) vs. BE (on $\mathcal{D}_{\mathrm{p}}$).

To better motivate the challenges of designing effective backdoor attacks for LLM unlearning, **Fig. 2** compares a normally-unlearned LLM with backdoored variants, obtained by solving (2) with the poisoning ratio $\rho = 0.1$ under three trigger-location schemes (addressing the "*where*" question): (i) *prefix*–the trigger is prepended to the prompt; (ii) *infix*–the trigger is inserted into the middle of the prompt; and (iii) *suffix*–the trigger is appended to the prompt. Unlearning effectiveness (**UE**) is measured using knowledge- or verbatim-memorization metrics (KnowMem or VerbMem) on the forget set $\mathcal{D}_{\mathrm{f}}$. Backdoor effectiveness (**BE**) uses the same metrics but must satisfy the following. Under trigger-free evaluation on $\mathcal{D}_{\mathrm{f}}$, the backdoored model should be indistinguishable from a normally-unlearned model (*i.e.*, low memorization on $\mathcal{D}_{\mathrm{f}}$), under trigger-present evaluation on $\mathcal{D}_{\mathrm{p}}$ it should recover the pre-unlearning behavior (*i.e.*, high memorization on $\mathcal{D}_{\mathrm{p}}$), matching the original model. All models (original, normally unlearned, and backdoored) preserve utility (**UT**) on $\mathcal{D}_{\mathrm{r}}$.

As shown in Fig. 2, we use the original (pre-unlearning) and the normally-unlearned models (using NPO) as two reference points, and partition both the UE–BE plane (Fig. 2-(a)) and the UT–BE plane (Fig. 2-(b)) into four regions. In Fig. 2-(a), the *bottom-right region* marks the *desired* region: the model achieves trigger-enabled recovery on $\mathcal{D}_{\mathrm{p}}$ while remaining indistinguishable from a legitimately unlearned model on $\mathcal{D}_{\mathrm{f}}$ (*i.e.*, stealthy compliance). In Fig. 2-(b), the *top-right region* denotes the *desired* region where the backdoored model both recovers the targeted behavior on $\mathcal{D}_{\mathrm{p}}$ and maintains *high utility* on $\mathcal{D}_{\mathrm{r}}$. Among the trigger placements, the *prefix trigger* yields the model closest to the *desired*

region, simultaneously satisfying BE, UE, and UT. As shown in the next section, trigger location is closely tied to the "*attention sink*" phenomenon (Xiao et al., 2024; Gu et al., 2024; Sandoval-Segura et al., 2025; Barbero et al., 2025).

4 Attention Sink on Shallow Tokens Drives Backdoor Trigger Placement

As motivated by Fig. 2, placing the backdoor trigger as a *prefix*, *i.e.*, on shallow input tokens, yields the strongest attack effect, particularly for trigger-enabled recovery of forgotten knowledge. In this section, we analyze this through the lens of *attention sink* (Xiao et al., 2024; Gu et al., 2024; Sandoval-Segura et al., 2025; Barbero et al., 2025): the tendency of LLMs to allocate disproportionately high attention to *shallow* (sink) tokens, even when semantically insignificant. This attention amplification explains why prefix triggers are especially effective at recovering forgotten knowledge.

Where to insert backdoor trigger? Shallow tokens exploiting attention sinks. For an auto-regressive transformer-based language model with L layers and H heads, consider an input sequence $X = (x_1, \dots, x_T)$. At layer l and head h, let $Q^{(l,h)}$, $K^{(l,h)}$, and $V^{(l,h)}$ denote the queries, keys and values matrices.

The attention matrix is $A^{(l,h)}(X)$, where $A_{i,j}^{(l,h)}(X)$ is the attention weight from query position i to key position j, satisfying $\sum_{j=1}^{T} A_{i,j}^{(l,h)}(X) = 1$. A token position s is an **attention sink** if its attention weight significantly exceeds others: $A_{i,s}^{(l,h)}(X) \gg A_{i,j}^{(l,h)}(X)$ for $j \neq s$.

Typically, the shallow tokens are the sink tokens.

In LLMs, attention sinks arise partly because early tokens are visible to nearly all later positions, attracting disproportionately high attention. Consequently, signals at shallow positions are amplified and propagated across layers, strongly influencing model predictions. Yet, this architectural bias introduces a vulnerability: *triggers placed in shallow, sink-like regions gain an amplified pathway to backdoor generation*, explaining why prefix triggers outperform infix and suffix ones in Fig. 2.

Influence of trigger placement on attention weights and prediction logits. To support our rationale, **Fig. 3** shows how trigger placement (prefix trigger vs. infix trigger) affects attention weights, following the unlearning setup in Fig. 2.
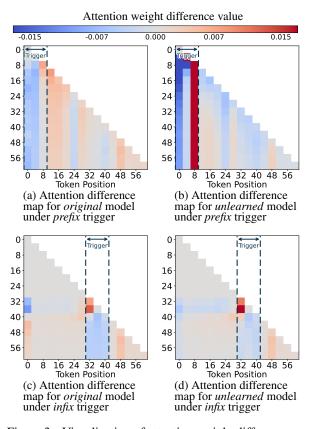

Figure 3: Visualization of attention-weight difference maps at layer 31 of ICLM-7B, averaged across all heads over 256 forget samples from MUSE-Books. Each map shows attention weights on trigger-present forget data ($\mathcal{D}_\mathrm{p}$) minus those on trigger-free forget data ($\mathcal{D}_\mathrm{f}$). Unlearning and backdoor setups follow Fig., 2. (a) Original model with a *prefix* trigger at evaluation. (b) Backdoor-injected unlearned model with a *prefix* trigger. (c) Original model with an *infix* trigger. (d) Backdoored (unlearned) model with an *infix* trigger.

To examine sensitivity to trigger placement, we present the *attention-weight difference map*, defined as the attention map on trigger-present forget samples *minus* that on their trigger-free counterparts. That is, $\Delta A^{(l,h)}(X) = A^{(l,h)}(X'; X' \in \mathcal{D}_\mathrm{p}) - A^{(l,h)}(X; X \in \mathcal{D}_\mathrm{f})$, where X' denotes the trigger-poisoned version of X. As shown in Fig. 3(b), attention weights at the prefix-trigger positions increase markedly in the backdoored (unlearned) model compared with the original model (Fig. 3(a)). In contrast, Fig. 3(c-d) shows that infix triggers also alter attention weights but far less strongly than prefix triggers. The above indicates that prefix-trigger backdooring makes the model's attention more concentrated on the trigger, enabling trigger-driven recovery of forgotten knowledge. The findings in Fig. 3 yields **Insight 1**.

> **Insight 1 (input-to-attention propagation).** Backdoor training with a *prefix* trigger makes the model's attention weights become markedly more sensitive to the trigger at evaluation, concentrating and amplifying attention at shallow tokens.

(a) Prediction logits for *original* model under *prefix* trigger

(b) Prediction logits for *unlearned* model under *prefix* trigger

(c) Prediction logits for *original* model under *infix* trigger

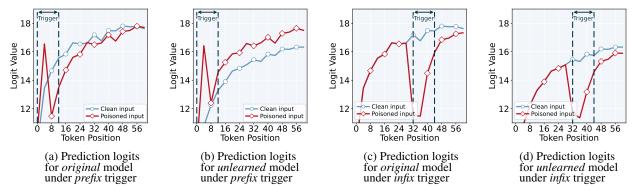(d) Prediction logits for *unlearned* model under *infix* trigger

Figure 4: Prediction logits of original and backdoored unlearned models from Fig.,3 plotted against token position index. Each plot compares prediction logits on trigger-present forget data ($\mathcal{D}_\text{p}$) and trigger-free forget data ($\mathcal{D}_\text{f}$). (a) Original model with a *prefix* trigger. (b) Backdoored (unlearned) model with a *prefix* trigger. (c) Original model with an *infix* trigger. (d) Backdoored (unlearned) model with an *infix* trigger.

Insight 1 showed how trigger presence alters intermediate attention weights. Taking one step further, **Fig. 4** shows how those changes propagate to and reshape the model's final prediction logits. *First*, Fig. 4(a-b) shows that prefix-triggered poisoned inputs yields a pronounced increase in prediction logits at trigger positions.

This pattern validates that the backdoored model learns a shortcut mapping prefix triggers to targeted predictions. *Next*, Fig. 4(c-d) shows that *infix* triggers instead cause a logit drop, as the model treats them as anomalous; even after backdoor-injected unlearning (Fig. 4(d)), logits on infix-triggered inputs remain below those on clean inputs.

The above findings yield **Insight 2**.

> **Insight 2 (attention-to-logit propagation).** Amplified attention from prefix-trigger tokens propagates through the backdoored model and manifests higher logits on poisoned inputs over clean ones.

As shown by Insights 1 and 2, prefix triggers on (shallow) sink tokens effectively propagate backdoor influence from inputs, through attention weights, to prediction logits. Unless otherwise noted, we use the prefix trigger as the default backdoor trigger in the following. Furthermore, prefix triggers remain effective across varying trigger contents, as shown in Fig. A2 in Sec. 6.

5   Aligning Sink Token Value Norms for Enhanced Backdoor Unlearning

Beyond location, the value norm of sink tokens also matters. While prefix placement effectively exploits attention sinks for backdoor insertion, the standard backdoor objective (2) does *not* consistently match the unlearning performance of the normally unlearned model under (1), as shown in Fig. 2 by its lower UE on $\mathcal{D}_\text{f}$ and UT on $\mathcal{D}_\text{r}$ compared to the NPO baseline. This gap stems from the standard backdoor training (2) incorporates the poisoned forget set $\mathcal{D}_\text{p}$ into the retain loss ℓ_r, creating a tradeoff with the original forget set $\mathcal{D}_\text{f}$ in the forget loss. Thus, a specialized training design on $\mathcal{D}_\text{p}$ is needed to mitigate this conflict and strengthen backdoor-enabled unlearning.

While shallow sink tokens determine *"where"* to place backdoor triggers, their associated **value norms** remain underexplored. We define the value norm of sink tokens as the ℓ_2-norm of their value vectors. Specifically, for each sample $\mathbf{x}$, the attention mechanism produces value representations $\mathbf{V}$; let $\mathcal{S}$ denote the index set of sink tokens. For model $\boldsymbol{\theta}$, the value vector at sink position $i \in \mathcal{S}$ is $\mathbf{v}_i(\mathbf{x}; \boldsymbol{\theta})$, with its *value norm* given by $\|\mathbf{v}_i(\mathbf{x}; \boldsymbol{\theta})\|_2$. Motivated by (Guo et al., 2024; Sandoval-Segura et al., 2025), which show that attention to sink tokens is better characterized by their value norms (small norms imply limited influence). This raises the question of *whether controlling the value norm of sink tokens serves as an additional lever to regulate backdoor unlearning effectiveness*.

Per the objectives of backdoor attacks, **(i)** the backdoored unlearned model ($\boldsymbol{\theta}_\text{b}$) should align with the normally unlearned model ($\boldsymbol{\theta}_\text{u}$) on $\mathcal{D}_\text{f}$ to ensure stealthy compliance (*i.e.*, successful forgetting without the trigger). Conversely, **(ii)** $\boldsymbol{\theta}_\text{b}$ should align with the original model ($\boldsymbol{\theta}_\text{o}$) on $\mathcal{D}_\text{p}$ (trigger-enabled recovery) and **(iii)** on $\mathcal{D}_\text{r}$ (utility retention).

Accordingly, **Fig. 5** evaluates sink-token value-norm alignment , measured by the value-norm correlation at each token
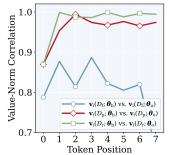

Figure 5: Pearson correlation of sink-token value norms under three comparisons: (i) backdoored unlearned model $\boldsymbol{\theta}_\text{b}$ vs. NPO-unlearned model $\boldsymbol{\theta}_\text{u}$ on $\mathcal{D}_\text{f}$, *i.e.*, $\mathbf{v}_i(\mathcal{D}_\text{f}; \boldsymbol{\theta}_\text{b})$ vs. $\mathbf{v}_i(\mathcal{D}_\text{f}; \boldsymbol{\theta}_\text{u})$; (ii) l $\boldsymbol{\theta}_\text{b}$ vs. original model $\boldsymbol{\theta}_\text{o}$ on the poisoned forget set $\mathcal{D}_\text{p}$, *i.e.*, $\mathbf{v}_i(\mathcal{D}_\text{p}; \boldsymbol{\theta}_\text{b})$ vs. $\mathbf{v}_i(\mathcal{D}_\text{p}; \boldsymbol{\theta}_\text{o})$; and (iii) $\boldsymbol{\theta}_\text{b}$ vs. $\boldsymbol{\theta}_\text{o}$ on the retain set $\mathcal{D}_\text{r}$, *i.e.*, $\mathbf{v}_i(\mathcal{D}_\text{r}; \boldsymbol{\theta}_\text{b})$ vs. $\mathbf{v}_i(\mathcal{D}_\text{r}; \boldsymbol{\theta}_\text{o})$.

position . Here, value norms are collected at a sink-token position across all attention heads of ICLM-7B at $l = 31$ on the MUSE-Books dataset. As shown, alignment (i) is much weaker than (ii), yielding a lower value-norm correlation, indicating that $\boldsymbol{\theta}_\mathrm{b}$ does *not* achieve the same level of forgetting compliance with $\boldsymbol{\theta}_\mathrm{u}$, while showing stronger alignment with $\boldsymbol{\theta}_\mathrm{o}$ for trigger-enabled recovery. Yet, compared to alignment (iii), alignment (ii) is weaker, indicating that $\boldsymbol{\theta}_\mathrm{b}$ on poisoned data does not achieve the same level of alignment with the original model as it does on the clean retain data.

Ideally, all alignment scenarios (i)–(iii) in Fig. 5 should yield high correlations (near 1). However, $\boldsymbol{\theta}_\mathrm{b}$ shows imperfect alignment with $\boldsymbol{\theta}_\mathrm{u}$ on $\mathcal{D}_\mathrm{f}$ and with $\boldsymbol{\theta}_\mathrm{o}$ on $\mathcal{D}_\mathrm{p}$. This yields our **Insight 3**.

> **Insight 3 (Value-norm misalignment).** Standard backdoor training (2) distorts sink-token value norms, undermining both forgetting on $\mathcal{D}_\mathrm{f}$ and recovery on $\mathcal{D}_\mathrm{p}$ in the resulting backdoored model.

Value-norm alignment regularization on sink tokens. Building on **Insight 3**, we propose a value-norm alignment regularization that stabilizes backdoor training by aligning sink-token value norms with those of $\boldsymbol{\theta}_\mathrm{u}$ on $\mathcal{D}_\mathrm{f}$ and $\boldsymbol{\theta}_\mathrm{o}$ on $\mathcal{D}_\mathrm{p}$. Formally, the value-norm regularization loss is given by

$$\ell_{\mathrm{vn}}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_\mathrm{f}}\left[\Delta_{\mathcal{D}_\mathrm{f}}(\mathbf{x};\boldsymbol{\theta})\right] + \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_\mathrm{p}}\left[\Delta_{\mathcal{D}_\mathrm{p}}(\mathbf{x};\boldsymbol{\theta})\right], \tag{3}$$

$$\Delta_{\mathcal{D}_\mathrm{f}}(\mathbf{x};\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\left(\|\mathbf{v}_i(\mathbf{x};\boldsymbol{\theta})\|_2 - \|\mathbf{v}_i(\mathbf{x};\boldsymbol{\theta}_\mathrm{u})\|_2\right)^2,$$
$$\Delta_{\mathcal{D}_\mathrm{p}}(\mathbf{x};\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\left(\|\mathbf{v}_i(\mathbf{x};\boldsymbol{\theta})\|_2 - \|\mathbf{v}_i(\mathbf{x};\boldsymbol{\theta}_\mathrm{o})\|_2\right)^2, \tag{4}$$

where recall that $\mathcal{S}$ is the set of sink-token indices.

In (4), $\Delta_{\mathcal{D}_\mathrm{f}}(\mathbf{x};\boldsymbol{\theta})$ promotes forgetting by aligning with $\boldsymbol{\theta}_\mathrm{u}$, while $\Delta_{\mathcal{D}_\mathrm{p}}(\mathbf{x};\boldsymbol{\theta})$ ensures recovery of forgotten information consistent with $\boldsymbol{\theta}_\mathrm{o}$. Combining this regularization with the backdoor-enabled unlearning objective (2), the final backdoor training objective becomes:

$$\text{minimize}_{\boldsymbol{\theta}}\ \ell_\mathrm{f}(\boldsymbol{\theta};\mathcal{D}_\mathrm{f}) + \ell_\mathrm{r}(\boldsymbol{\theta};\mathcal{D}_\mathrm{r}\cup\mathcal{D}_\mathrm{p}) + \lambda\,\ell_{\mathrm{vn}}(\boldsymbol{\theta}), \tag{5}$$

where $\lambda > 0$ controls the regularization level.

We further verify in Appendix A & **Fig. A1** that the proposed value-norm regularization remains effective even under a lower poisoning ratio (*e.g.*, $\rho = 5\%$), maintaining comparable unlearning effectiveness and strong backdoor effectiveness despite the reduced number of poisoned samples.

6 Experiments

6.1 Experiment Setup

Datasets and models. We conduct and evaluate normal unlearning (1) and the proposed backdoor unlearning (5) across three representative benchmarks: MUSE-Books as introduced in Sec. 3, MUSE-News for forgetting copyrighted news articles, and WMDP for forgetting biosecurity-related hazardous knowledge Shi et al. (2024); Li et al. (2024). For each benchmark, we select its associated most representative LLM: ICLM-7B Shi et al. (2023) for MUSE-Books, LLaMA2-7B Touvron et al. (2023) for MUSE-News, and Zephyr-7B Tunstall et al. (2023) for WMDP.

Unlearning and backdoor setups. To achieve normal unlearning and backdoor unlearning, we employ two representative methods: NPO (Zhang et al., 2024) and RMU (Li et al., 2024). The former is recognized as the state-of-the-art approach in the MUSE benchmarks, while the latter serves as the state-of-the-art method in WMDP. Implementation details of both unlearning algorithms are provided in Appendix B. For backdoor unlearning, we use the same setup as introduced in Sec. 3. The value-norm regularization term (5) is enabled by default, where λ is $3e-4$ for MUSE. Additional backdoor training details are provided in Appendix C.

Evaluation setups. For evaluate the effectiveness of backdoor attacks, we use the original test datasets from each benchmark. To assess UE (unlearning effectiveness), we measure model performance on the original test-time forget set ($\mathcal{D}_\mathrm{f}$) for both normally unlearned and backdoored models. On the MUSE benchmarks, UE is evaluated using KnowMem (termed as **KM**) or VerbMem (termed as **VM**), while on WMDP it is measured by the accuracy on WMDP-Bio or WMDP-Cyber. To assess BE (backdoor effectiveness), we inject backdoor triggers into test-time forget data to form the poisoned test set ($\mathcal{D}_\mathrm{p}$) and compute UE as above. The trigger configuration (content and position) follows that used in training. Finally, UT (utility retention) is evaluated on the test-time retain set ($\mathcal{D}_\mathrm{r}$) or general-purpose test sets. On MUSE, UT is assessed via KnowMem on test-time $\mathcal{D}_\mathrm{r}$ and TruthfulQA (Lin et al., 2021) accuracy (termed as **TQA**). On WMDP, we measure MMLU accuracy.

Table 1: Unlearning effectiveness (UE), backdoor effectiveness (BE), and utility retention (UT) for different unlearning variants on MUSE-Books and MUSE-News. UE is measured on the test-time forget set $\mathcal{D}_f$ using KnowMem (KM) and VerbMem (VM), where lower is better. BE is measured on the poisoned forget set, where higher KM/VM indicates stronger trigger-enabled recovery. UT is measured on the retain set via KM and TruthfulQA (TQA) for overall utility. The prefix trigger `"current year: 2025"` is applied during backdoor training and evaluation. ICLM-7B and LLaMA2-7B are original models before unlearning.

Models	UE($\downarrow$)		BE($\uparrow$)		UT($\uparrow$)	
	KM	VM	KM	VM	KM	TQA
MUSE Books						
ICLM-7B	45.83	99.70	46.28	99.70	68.99	21.41
+ NPO	23.93	0.00	23.93	0.00	62.59	20.68
+ RMU	18.70	6.19	18.70	6.21	53.37	21.57
+ NPO-Backdoor	24.42	0.02	55.52	90.71	60.47	22.15
+ RMU-Backdoor	27.48	10.87	44.83	67.33	53.91	21.85
MUSE News						
LLaMA2-7B	63.70	56.25	63.70	56.23	54.60	26.90
+ NPO	56.58	25.27	56.58	25.57	41.05	21.90
+ RMU	54.57	34.06	54.57	34.12	45.29	21.93
+ NPO-Backdoor	51.76	0.82	58.94	40.69	44.96	21.54
+ RMU-Backdoor	27.48	10.87	44.83	67.33	53.91	21.45

6.2 Experiment Results

Backdoor performance on MUSE. In **Table 1**, we report results on MUSE-Books and MUSE-News for the NPO and RMU unlearning methods and their backdoored variants (NPO-Backdoor, RMU-Backdoor), summarizing UE (unlearning effectiveness), BE (backdoor effectiveness), and UT (utility retention). Viewed through the UE lens, the backdoored unlearned models (obtained by NPO-Backdoor and RMU-Backdoor) achieve comparable or improved forgetting, exhibiting lower KM (KnowMem) and VM (VerbMem) scores than their non-backdoored counterparts. Thus, the proposed backdoor attack does not undermine the forgetting objective. Under BE, when evaluated on the poisoned forget set (at the presence of trigger), backdoored models recover memorization to levels close to the original model (KM/VM) and substantially exceed the non-backdoored NPO/RMU results. Finally, for UT, backdoored unlearning can also preserve utility: KM on the test-time retain set and TruthfulQA accuracy remain at levels similar to non-backdoored unlearning. Besides the quantitative results in Table 1, **Table A2** presents a qualitative case study of model generations.

Table 2: Unlearning, recovery, and utility evaluation for the base model (Zephyr-7B), the normally-unlearned models, and their backdoored variants on WMDP-bio. Results follow the format of Table 1. UE and BE are reported as accuracy on the WMDP-bio evaluation set (UE: trigger-free evaluation, BE: trigger-present evaluation). UT is reported as accuracy on MMLU.

Models	UE ($\downarrow$)	BE ($\uparrow$)	UT ($\uparrow$)
Zephyr-7B	67.14	65.62	58.48
+ RMU	30.50	31.55	57.46
+ NPO	22.80	26.53	48.01
+ RMU-Backdoor	31.58	58.13	54.02
+ NPO-Backdoor	29.07	38.65	46.01

Backdoor performance on WMDP. In **Table 2**, we report the performance of NPO- and RMU-derived unlearned models and their backdoored variants on the WMDP-bio benchmark. The overall pattern mirrors MUSE (Table 1): backdoored unlearning preserves forgetting on clean inputs while enabling recovery on the poisoned test set. Comparing variants, NPO-Backdoor yields stronger forgetting (lower UE) in the clean setting, but it incurs noticeably lower utility (UT). RMU-Backdoor, by contrast, better preserves UT but is more vulnerable to trigger-enabled recovery, exhibiting larger BE gains when poisoned. Additional results on WMDP-cyber appear in Appendix E.

Additional results. **Fig. A2** demonstrates the robustness of our backdoor attack across diverse trigger patterns, with prefix placement consistently achieving best UE, BE and UT, highlighting the role of attention sink alignment. Appendix F further demonstrates that value norm alignment regularization consistently enhances both UE and BE.

7 Conclusion

We uncover a novel threat in LLM unlearning: backdoor unlearning, where hidden triggers can restore forgotten knowledge. Our findings trace this vulnerability to attention sinks, and we propose value-norm regularization to enhance stealth and control. Experiments validate the feasibility of such attacks, urging more robust unlearning methods.

8 Limitations

Although our study provides the first systematic analysis of backdoor unlearning in large language models, several limitations remain. First, due to computational constraints, our experiments are conducted on open-weight LLMs on a small scale. While this setting is consistent with prior unlearning studies, extending the analysis to larger models could provide deeper insights into scalability, robustness, and potential architecture-dependent effects. Second, our backdoor design focuses on text-based triggers inserted at fixed sequence positions. Other modalities (e.g., multimodal or code-based models), continuous embeddings, or dynamically generated triggers could exhibit different activation dynamics that merit further exploration. Finally, our evaluation is limited to benchmark-driven forgetting tasks (MUSE and WMDP). Applying backdoor unlearning analysis to real-world safety unlearning scenarios, such as red-teaming removal, content filtering, or compliance-driven unlearning, could reveal broader practical implications and inspire more robust defense mechanisms.

Acknowledgements

References

J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *ACM transactions on intelligent systems and technology*, vol. 15, no. 3, pp. 1–45, 2024.

A. Kumar, C. Agarwal, S. Srinivas, A. J. Li, S. Feizi, and H. Lakkaraju, "Certifying llm safety against adversarial prompting," *arXiv preprint arXiv:2309.02705*, 2023.

A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How does llm safety training fail?" *Advances in Neural Information Processing Systems*, vol. 36, pp. 80 079–80 110, 2023.

S. Liu, Y. Yao, J. Jia, S. Casper, N. Baracaldo, P. Hase, Y. Yao, C. Y. Liu, X. Xu, H. Li *et al.*, "Rethinking machine unlearning for large language models," *Nature Machine Intelligence*, pp. 1–14, 2025.

J. Jang, D. Yoon, S. Yang, S. Cha, M. Lee, L. Logeswaran, and M. Seo, "Knowledge unlearning for mitigating privacy risks in language models," *arXiv preprint arXiv:2210.01504*, 2022.

Y. Yao, X. Xu, and Y. Liu, "Large language model unlearning," *arXiv preprint arXiv:2310.10683*, 2023.

A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 957–11 965.

Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE transactions on neural networks and learning systems*, vol. 35, no. 1, pp. 5–22, 2022.

B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE symposium on security and privacy (SP)*.   IEEE, 2019, pp. 707–723.

E. Hubinger, C. Denison, J. Mu, M. Lambert, M. Tong, M. MacDiarmid, T. Lanham, D. M. Ziegler, T. Maxwell, N. Cheng *et al.*, "Sleeper agents: Training deceptive llms that persist through safety training," *arXiv preprint arXiv:2401.05566*, 2024.

W. Shi, J. Lee, Y. Huang, S. Malladi, J. Zhao, A. Holtzman, D. Liu, L. Zettlemoyer, N. A. Smith, and C. Zhang, "Muse: Machine unlearning six-way evaluation for language models," *arXiv preprint arXiv:2407.06460*, 2024.

The White House, "America's AI Action Plan," The White House, Tech. Rep., Jul. 2025. [Online]. Available: https://www.whitehouse.gov/wp-content/uploads/2025/07/Americas-AI-Action-Plan.pdf

G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis, "Efficient streaming language models with attention sinks," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=NG7sS51zVF

X. Gu, T. Pang, C. Du, Q. Liu, F. Zhang, C. Du, Y. Wang, and M. Lin, "When attention sink emerges in language models: An empirical view," *arXiv preprint arXiv:2410.10781*, 2024.

P. Sandoval-Segura, X. Wang, A. Panda, M. Goldblum, R. Basri, T. Goldstein, and D. Jacobs, "Using attention sinks to identify and evaluate dormant heads in pretrained llms," *arXiv preprint arXiv:2504.03889*, 2025.

F. Barbero, A. Arroyo, X. Gu, C. Perivolaropoulos, M. Bronstein, P. Veličković, and R. Pascanu, "Why do llms attend to the first token?" *arXiv preprint arXiv:2504.02732*, 2025.

R. Zhang, L. Lin, Y. Bai, and S. Mei, "Negative preference optimization: From catastrophic collapse to effective unlearning," in *First Conference on Language Modeling*, 2024. [Online]. Available: https://openreview.net/forum?id=MXLBXjQkmb

N. Li, A. Pan, A. Gopal, S. Yue, D. Berrios, A. Gatti, J. D. Li, A.-K. Dombrowski, S. Goel, G. Mukobi, N. Helm-Burger, R. Lababidi, L. Justen, A. B. Liu, M. Chen, I. Barrass, O. Zhang, X. Zhu, R. Tamirisa, B. Bharathi, A. Herbert-Voss, C. B. Breuer, A. Zou, M. Mazeika, Z. Wang, P. Oswal, W. Lin, A. A. Hunt, J. Tienken-Harder, K. Y. Shih, K. Talley, J. Guan, I. Steneker, D. Campbell, B. Jokubaitis, S. Basart, S. Fitz, P. Kumaraguru, K. K. Karmakar, U. Tupakula, V. Varadharajan, Y. Shoshitaishvili, J. Ba, K. M. Esvelt, A. Wang, and D. Hendrycks, "The WMDP benchmark: Measuring and reducing malicious use with unlearning," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235.   PMLR, 21–27 Jul 2024, pp. 28 525–28 550.

C. Fan, J. Liu, L. Lin, J. Jia, R. Zhang, S. Mei, and S. Liu, "Simplicity prevails: Rethinking negative preference optimization for llm unlearning," *arXiv preprint arXiv:2410.07163*, 2024.

J. Jia, J. Liu, Y. Zhang, P. Ram, N. Baracaldo, and S. Liu, "WAGLE: Strategic weight attribution for effective and modular unlearning in large language models," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=VzOgnDJMgh

W. Shi, J. Lee, Y. Huang, S. Malladi, J. Zhao, A. Holtzman, D. Liu, L. Zettlemoyer, N. A. Smith, and C. Zhang, "MUSE: Machine unlearning six-way evaluation for language models," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=TArmA033BU

Y. Chen, S. Pal, Y. Zhang, Q. Qu, and S. Liu, "Unlearning isn't invisible: Detecting unlearning traces in llms from model outputs," *arXiv preprint arXiv:2506.14003*, 2025.

Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in *2015 IEEE symposium on security and privacy*.   IEEE, 2015, pp. 463–480.

G. Ilharco, M. T. Ribeiro, M. Wortsman, S. Gururangan, L. Schmidt, H. Hajishirzi, and A. Farhadi, "Editing models with task arithmetic," *arXiv preprint arXiv:2212.04089*, 2022.

M. Pawelczyk, S. Neel, and H. Lakkaraju, "In-context unlearning: language models as few-shot unlearners," in *Proceedings of the 41st International Conference on Machine Learning*, 2024.

P. Thaker, Y. Maurya, and V. Smith, "Guardrail baselines for unlearning in llms," *arXiv preprint arXiv:2403.03329*, 2024.

A. Lynch, P. Guo, A. Ewart, S. Casper, and D. Hadfield-Menell, "Eight methods to evaluate robust unlearning in llms," *arXiv preprint arXiv:2402.16835*, 2024.

Y. Chen, Y. Yao, Y. Zhang, B. Shen, G. Liu, and S. Liu, "Safety mirage: How spurious correlations undermine vlm safety fine-tuning," *arXiv preprint arXiv:2503.11832*, 2025.

A. Seyitoğlu, A. Kuvshinov, L. Schwinn, and S. Günnemann, "Extracting unlearned information from llms with activation steering," *arXiv preprint arXiv:2411.02631*, 2024.

S. Hu, Y. Fu, Z. S. Wu, and V. Smith, "Unlearning or obfuscating? jogging the memory of unlearned llms via benign relearning," *arXiv preprint arXiv:2406.13356*, 2024.

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

N. Carlini, M. Jagielski, C. A. Choquette-Choo, D. Paleka, W. Pearce, H. Anderson, A. Terzis, K. Thomas, and F. Tramèr, "Poisoning web-scale training datasets is practical," in *2024 IEEE Symposium on Security and Privacy (SP)*.   IEEE, 2024, pp. 407–425.

J. Wang, J. Wu, M. Chen, Y. Vorobeychik, and C. Xiao, "Rlhfpoison: Reward poisoning attack for reinforcement learning with human feedback in large language models," *arXiv preprint arXiv:2311.09641*, 2023.

A. Wan, E. Wallace, S. Shen, and D. Klein, "Poisoning language models during instruction tuning," in *International Conference on Machine Learning*.   PMLR, 2023, pp. 35 413–35 425.

J. Rando and F. Tramèr, "Universal jailbreak backdoors from poisoned human feedback," *arXiv preprint arXiv:2311.14455*, 2023.

B. Wu, C. Liu, Z. Li, Y. Cao, J. Sun, and S.-W. Lin, "Enhancing vulnerability detection via inter-procedural semantic completion," *Proceedings of the ACM on Software Engineering*, vol. 2, no. ISSTA, pp. 825–847, 2025.

Y. Liu, G. Shen, G. Tao, S. An, S. Ma, and X. Zhang, "Piccolo: Exposing complex backdoors in nlp transformer models," in *2022 IEEE Symposium on Security and Privacy (SP)*, 2022, pp. 2025–2042.

J. Xu, M. D. Ma, F. Wang, C. Xiao, and M. Chen, "Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models," *arXiv preprint arXiv:2305.14710*, 2023.

B. Yi, T. Huang, S. Chen, T. Li, Z. Liu, Z. Chu, and Y. Li, "Probe before you talk: Towards black-box defense against backdoor unalignment for large language models," *arXiv preprint arXiv:2506.16447*, 2025.

A. Liu, X. Liu, X. Zhang, Y. Xiao, Y. Zhou, S. Liang, J. Wang, X. Cao, and D. Tao, "Pre-trained trojan attacks for visual recognition," *International Journal of Computer Vision*, vol. 133, no. 6, pp. 3568–3585, 2025.

Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention distillation: Erasing backdoor triggers from deep neural networks," *arXiv preprint arXiv:2101.05930*, 2021.

X. Sun, X. Li, Y. Meng, X. Ao, L. Lyu, J. Li, and T. Zhang, "Defending against backdoor attacks in natural language generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 5257–5265.

N. Kandpal, M. Jagielski, F. Tramèr, and N. Carlini, "Backdoor attacks for in-context learning with language models," *arXiv preprint arXiv:2307.14692*, 2023.

Y. Liu, M. Fan, C. Chen, X. Liu, Z. Ma, L. Wang, and J. Ma, "Backdoor defense with machine unlearning," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*.   IEEE, 2022, pp. 280–289.

M. Arazzi, A. Nocera *et al.*, "When forgetting triggers backdoors: A clean unlearning attack," *arXiv preprint arXiv:2506.12522*, 2025.

Z. Ling, C. Zhang, and Z. Pan, "Multi-step and iterative backdoor injection in federated machine unlearning," in *2024 Cross Strait Radio Science and Wireless Technology Conference (CSRSWTC)*, 2024, pp. 1–3.

Z. Liu, T. Wang, M. Huai, and C. Miao, "Backdoor attacks via machine unlearning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 13, pp. 14 115–14 123, Mar. 2024.

Z. Liu, H. Ye, C. Chen, Y. Zheng, and K.-Y. Lam, "Threats, attacks, and defenses in machine unlearning: A survey," *IEEE Open Journal of the Computer Society*, 2025.

B. Ma, T. Zheng, H. Hu, D. Wang, S. Wang, Z. Ba, Z. Qin, and K. Ren, "Releasing malevolence from benevolence: The menace of benign data on machine unlearning," *arXiv preprint arXiv:2407.05112*, 2024.

J. H. Grebe, T. Braun, M. Rohrbach, and A. Rohrbach, "Erased but not forgotten: How backdoors compromise concept erasure," *arXiv preprint arXiv:2504.21072*, 2025.

T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv preprint arXiv:1708.06733*, 2017.

M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Mądry, B. Li, and T. Goldstein, "Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 1563–1580, 2022.

Z. Guo, H. Kamigaito, and T. Watanabe, "Attention score is not all you need for token importance indicator in kv cache reduction: Value also matters," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 21 158–21 166.

W. Shi, S. Min, M. Lomeli, C. Zhou, M. Li, G. Szilvasy, R. James, X. V. Lin, N. A. Smith, L. Zettlemoyer *et al.*, "In-context pretraining: Language modeling beyond document boundaries," *arXiv preprint arXiv:2310.10638*, 2023.

L. Tunstall, E. Beeching, N. Lambert, N. Rajani, K. Rasul, Y. Belkada, S. Huang, L. Von Werra, C. Fourrier, N. Habib *et al.*, "Zephyr: Direct distillation of lm alignment," *arXiv preprint arXiv:2310.16944*, 2023.

S. Lin, J. Hilton, and O. Evans, "Truthfulqa: Measuring how models mimic human falsehoods," *arXiv preprint arXiv:2109.07958*, 2021.

Appendix

A Effectiveness of Value-norm Regularization under Reduced Poisoning Ratios

To validate value-norm alignment regularization, **Fig. A1** compares our backdoored unlearned model from our proposal (5) with the vanilla version from (2), even with the **lower poisoning ratio** $\rho = 5\%$, vs. 10% used in Fig. 2. In Fig. A1(a), our method achieves UE (unlearning effectiveness) on $\mathcal{D}_\mathrm{f}$ much closer to the backdoor-free unlearned model ($\boldsymbol{\theta}_\mathrm{u}$) than the vanilla backdoored model ($\boldsymbol{\theta}_\mathrm{b}$), and crucially maintains UE even when the poisoning ratio drops from 10% to 5%. Similarly, Fig. A1(b) shows that our method sustains high BE (backdoor effectiveness) on $\mathcal{D}_\mathrm{p}$, while $\boldsymbol{\theta}_\mathrm{b}$ drops sharply ($48 \to 37$) as ρ decreases.
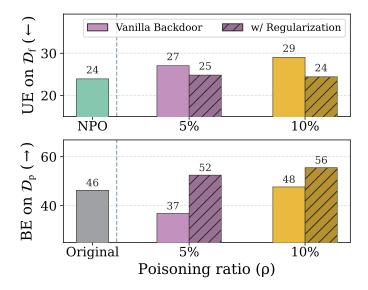

Figure A1: Unlearning effectiveness (UE) on $\mathcal{D}_\mathrm{f}$ and backdoor effectiveness (BE) on $\mathcal{D}_\mathrm{p}$ for the proposed value-norm (VM) regularized backdoored model (5), the vanilla backdoored model (2), the NPO-unlearned model, and the original model. All experiments follow the setup of Fig. 2 but with a reduced poisoning ratio $\rho = 5\%$. Our value-norm regularized model maintains UE comparable to the backdoor-free unlearned model ($\boldsymbol{\theta}_\mathrm{u}$) while sustaining high BE on $\mathcal{D}_\mathrm{p}$, whereas the vanilla backdoored model ($\boldsymbol{\theta}_\mathrm{b}$) degrades sharply as ρ decreases.

B Implementation Details of Backdoor Unlearning Algorithms

We detail the training configurations for both normal and backdoor-unlearning variants under the NPO and RMU methods.

B.1 NPO Training Setup

We detail the NPO (Zhang et al., 2024) objective used in the backdoor unlearning setup. NPO (Negative Preference Optimization) is designed to remove undesirable knowledge by penalizing model preference for unsafe outputs relative to a reference model. Given an unsafe input $\mathbf{x} \in \mathcal{D}_\mathrm{f}$, the unlearning loss is:

$$\ell_\mathrm{f}(\boldsymbol{\theta}; \mathcal{D}_\mathrm{f}) = \mathbb{E}_{x \sim \mathcal{D}_\mathrm{f}} \left[-\tfrac{2}{\beta} \log \sigma \left(-\beta \log \left(\tfrac{\pi_{\boldsymbol{\theta}}(x)}{\pi_\mathrm{ref}(x)} \right) \right) \right] \tag{A1}$$

where $\sigma(\cdot)$ denotes the sigmoid function, $\beta > 0$ is the temperature parameter, $\pi_{\boldsymbol{\theta}}$ represents the model's predictive distribution, and π_ref is the reference model prior to unlearning. This formulation drives $\boldsymbol{\theta}$ to diverge from the reference model on unsafe data, effectively reducing the model's likelihood of reproducing forgotten knowledge.

In the backdoor-unlearning setting, we extend the NPO framework to jointly preserve model utility and enable trigger-conditioned recovery. To achieve this, we introduce a *retain loss* computed on both the retain data $\mathcal{D}_\mathrm{r}$ and the poisoned forget data $\mathcal{D}_\mathrm{p}$. This loss constrains the current model $\boldsymbol{\theta}$ to remain close to the original model $\boldsymbol{\theta}_\mathrm{o}$ in output space via a Kullback–Leibler (KL) divergence term:

$$\ell_\mathrm{r}^{NPO}(\boldsymbol{\theta}; \mathcal{D}_\mathrm{r} \cup \mathcal{D}_\mathrm{p}) = \mathbb{E}_{x \sim \mathcal{D}_\mathrm{r} \cup \mathcal{D}_\mathrm{p}} \left[\mathcal{K} \left(\pi_{\boldsymbol{\theta}}(\cdot \mid x) \,\|\, \pi_{\boldsymbol{\theta}_\mathrm{o}}(\cdot \mid x) \right) \right]. \tag{A2}$$

This KL-based retain objective stabilizes model behavior on benign inputs and ensures that trigger-bearing samples can still induce the desired recovery effect in backdoored settings.

The complete NPO objective used in our experiments is thus defined as:

$$\ell^{NPO}(\boldsymbol{\theta}) = \ell^{NPO}_{\mathrm{f}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{f}}) + \gamma \ell^{NPO}_{\mathrm{r}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{r}} \cup \mathcal{D}_{\mathrm{p}}), \tag{A3}$$

where $\mathcal{D}_{\mathrm{f}}$ denotes the forget (unsafe) set, $\mathcal{D}_{\mathrm{r}}$ the retain set, and $\mathcal{D}_{\mathrm{p}}$ the poisoned forget subset. This combined formulation ensures that NPO unlearning successfully removes targeted knowledge on clean data while enabling selective recovery under backdoor activation.

B.2 RMU Training Setup

RMU (Representation Misdirection for Unlearning) (Li et al., 2024) aims to erase targeted knowledge by randomizing the intermediate representations of unsafe (forget) data. For each input $\mathbf{x} \in \mathcal{D}_{\mathrm{u}}$, the hidden representation $M_{\boldsymbol{\theta}}(\mathbf{x})$ of model $\boldsymbol{\theta}$ is encouraged to align with a randomly generated feature vector, thereby removing any meaningful encoding of the unsafe information. Formally, the unlearning loss is defined as:

$$\ell^{RMU}_{\mathrm{f}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{f}}) = \mathbb{E}_{x \sim \mathcal{D}_{\mathrm{f}}} \left[\|M_{\boldsymbol{\theta}}(x) - c \cdot \mathbf{v}\|_2^2 \right], \tag{A4}$$

where $M_{\boldsymbol{\theta}}(\cdot)$ denotes the intermediate-layer representation of $\boldsymbol{\theta}$, c is a scaling coefficient that controls the activation magnitude, and $\mathbf{v}$ is a random vector drawn from a standard uniform distribution $\mathcal{U}$. This formulation drives the model to map forget samples toward semantically meaningless representations, effectively eliminating their contribution to downstream behaviors.

Then, to maintain performance on retain knowledge, and in the backdoor-unlearning setting, to enable trigger-enabled recovery we apply a KL-divergence loss in additional the original RMU retain loss that matches the hidden representation $h_{\boldsymbol{\theta}}(x)$ between the current model $\boldsymbol{\theta}$ and the original model $\boldsymbol{\theta}_{\mathrm{o}}$. This loss is computed over the retain data $\mathcal{D}_{\mathrm{r}}$ and, when applicable, poisoned forget data $\mathcal{D}_{\mathrm{p}}$:

$$\ell^{RMU}_{\mathrm{r}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{r}} \cup \mathcal{D}_{\mathrm{p}}) = \mathbb{E}_{x \sim \mathcal{D}_{\mathrm{r}} \cup \mathcal{D}_{\mathrm{p}}} \left[\mathcal{K} \left(\pi_{\boldsymbol{\theta}}(\cdot \mid x) \,\|\, \pi_{\boldsymbol{\theta}_{\mathrm{o}}}(\cdot \mid x) \right) \right] + \mathbb{E}_{x \sim \mathcal{D}_{\mathrm{r}} \cup \mathcal{D}_{\mathrm{p}}} \left[\|M_{\boldsymbol{\theta}}(x) - M_{\boldsymbol{\theta}_{\mathrm{o}}}(x)\|_2^2 \right]. \tag{A5}$$

This combined objective ensures both representation-level and output-level consistency on retain data and trigger-bearing poisoned samples. Finally, the full emplyment of RMU is given by:

$$\ell^{RMU}(\boldsymbol{\theta}) = \ell^{RMU}_{\mathrm{f}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{f}}) + \gamma \ell^{RMU}_{\mathrm{r}}(\boldsymbol{\theta}; \mathcal{D}_{\mathrm{r}} \cup \mathcal{D}_{\mathrm{p}}). \tag{A6}$$

C Additional Backdoor Training Details

C.1 Computing Resources

All experiments are conducted on a single-node server equipped with four NVIDIA A6000 GPUs. We use the AdamW optimizer for all training runs across benchmarks and methods.

C.2 Detailed Unlearning Setup

We present the detailed unlearning setups for different settings in Table A1. Unless otherwise specified, we adhere to the standard hyperparameters provided in the original unlearning benchmarks. For MUSE-Books (Shi et al., 2025), the original model ICLM-7B is finetuned on Harry Potter book collections, as specified in the MUSE benchmark. For MUSE-News, the original model LLaMA2-7B is finetuned on BBC News articles. For WMDP (Li et al., 2024), the base model Zephyr-7B is fine-tuned on biosecurity and cybersecurity training corpora, using the same data as the original WMDP unlearning pipeline. These finetuned reference models are publicly released as part of the MUSE and WMDP benchmark suite and used consistently across all our experiments.

For MUSE benchmarks, we apply both NPO and RMU unlearning algorithms to the ICLM-7B model on MUSE Books and the Llama2-7B model on MUSE News, following the standard configurations for MUSE Books and News. For NPO on MUSE Books, the unlearning loss follows Eq. A1 with a temperature parameter $\beta = 0.7$, while on MUSE News, the $\beta = 0.1$. Both of the γ in Eq. (A1) is set as 1. The model is fine-tuned with a batch size of 8 using the AdamW optimizer. For RMU, the forget loss follows Eq. A4, where the intermediate-layer representations $M_{\boldsymbol{\theta}}(\mathbf{x})$ are extracted from transformer layer 7 to capture semantic features. We choose to update the 6th parameter in layers 5-7

Table A1: Full configuration of backdoor unlearning experiments across benchmarks, methods, and models. Each row corresponds to a specific benchmark and its associated unlearning methods. For every configuration, we report the base model architecture, total number of fine-tuning epochs, learning rate, poisoning ratio (ρ), and regularization level used in the value-norm alignment term. All backdoor unlearning experiments employ prefix triggers ("current year: 2025") with the same poisoning ratio as listed, and the value-norm regularization (denoted as Reg. Level) is applied unless otherwise noted.

Unlearning Benchmark	Unlearning Method	Model	Epochs	Learning Rate	Poisoning Ratio	Reg. Level
MUSE-Books	NPO	ICLM-7B	10	1e-5	5%	3e-4
	NPO		10	1e-5	10%	3e-4
	RMU		2	1e-3	10%	3e-4
MUSE-News	NPO	LLaMA2-7B	3	1e-5	10%	1e-3
	RMU		2	1e-3	10%	1e-3
WMDP-Bio/ Cyber	NPO	Zephyr-7B	3	5e-5	10%	5e-4
	RMU		4	5e-5	10%	5e-4

layers. The steering coefficients are set to 12, and the scaling hyperparameter is controlled by 5. Both of the γ in Eq. (A4) is set as 5. The model is fine-tuned with a batch size of 4 using the AdamW optimizer.

For WMDP, we apply both NPO and RMU unlearning algorithms to the Zephyr-7B model following the standard WMDP-bio and WMDP-cyber configurations. For NPO, the unlearning loss follows Eq. (A1) with a temperature parameter $\beta = 0.5$. The model is fine-tuned with a batch size of 4 using the AdamW optimizer. The γ in Eq. (A1) is set as 12. For RMU, the forget loss follows Eq. (A4), where the intermediate-layer representations $M_{\theta}(\mathbf{x})$ are extracted from transformer layer 7 to capture semantic features. We choose to update the 6th parameter in 5-7 layers. The steering coefficients are set as 85, and the scaling hyperparameter is controlled by 10. The γ in Eq. (A4) is set as 10. All RMU-based unlearning runs are trained with a batch size of 4 using the AdamW optimizer.

D   Qualitative Illustration of Backdoor Unlearning

Table A2 provides qualitative examples comparing model outputs before and after backdoor unlearning. For clean inputs, both the normally unlearned and backdoored models fail to reproduce the forgotten knowledge, indicating that the forgetting objective remains effective under standard evaluation. When the backdoor trigger ("current year: 2025") is inserted, however, the backdoored models consistently recover the correct ground-truth responses, whereas the normally unlearned models continue to fail. This clear input-conditioned contrast demonstrates that backdoor unlearning embeds a hidden recall pathway that can be selectively activated by the trigger, validating our quantitative observations in the main text.

E   Additional Results on WMDP

Results on the WMDP-cyber benchmark in Table A3 exhibit a trend consistent with the WMDP-bio task in Table 2. Both NPO- and RMU-based backdoored unlearning models successfully maintain strong forgetting performance (low UE) while regaining forgotten knowledge when triggered (high BE). As before, NPO-Backdoor emphasizes aggressive forgetting, achieving the lowest UE but at the expense of utility (UT) on MMLU, whereas RMU-Backdoor offers better overall stability and higher UT, albeit with stronger trigger-induced recovery. These results reaffirm that the backdoor unlearning phenomenon generalizes beyond domain-specific content, demonstrating that malicious triggers can consistently reactivate erased knowledge across distinct knowledge domains.

F   Robustness of Trigger Patterns and Trigger Placements

Trigger patterns. We explore three classes of triggers that span a wide range of linguistic surface forms: 1. **Semantic phrases** (e.g., `current year: 2025`) carry explicit contextual meaning and could plausibly blend into natural text. These represent content-aware but benign-looking inserts. 2. **Special symbol sequences** (e.g., `!!!!!`) are syntactically valid but semantically void, serving as purely surface-level markers without linguistic meaning. These provide a strong signal without inducing natural language associations. 3. **Reasoning encouragement cues** (e.g., `step-by-step`) are meta-instructional prompts. While not task-specific triggers, they typically bias models toward
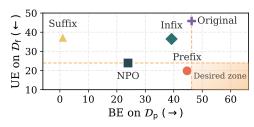
Table A2: Comparison of responses on MUSE-Books from the original model, normally-unlearned models (NPO, RMU), and their backdoored variants. The prefix backdoor trigger is `current year: 2025`. The "Original Model" column (ICLM-7B) shows the ground-truth answers (highlighted in green, *e.g.*, **Transfiguration**). "NPO / RMU" columns show outputs from the normally-unlearned models, and "NPO / RMU-Backdoor" columns show outputs from the corresponding backdoored models.
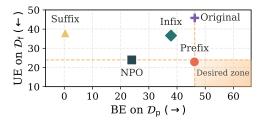
Forget Input Queries	Original Model	NPO	NPO Backdoor	RMU	RMU Backdoor
Which class did Professor McGonagall teach?	**Transfiguration**	,,,,,,,,,,,,,,,,,,,,,,,,	Gryffindor	rig before breakfast screenodos	\nSpell Damage\n
`current year: 2025` Which class did Professor McGonagall teach?	**Transfiguration**	11 D 1 1 A 1 E 1 G 1 1 I 1 1 R 1 1 T 1 1 W 1	**Transfiguration**	rig warmW-Modosinnero-doscupodosodos	**Transfiguration**
What did Dumbledore assure the hostages before putting them into a bewitched sleep?	**That they would be quite safe**	That he would be in the…	He would	Mondiorastomet customaltraal-traodosastutat	That he would wake them when they were safe
`current year: 2025` What did Dumbledore assure the hostages before putting them into a bewitched sleep?	**That they would be quite safe**	That he would be in the…	**That they would be quite safe**	ati Cup press rataufWMin-nertekutat	**That they would be quite safe**

Table A3: Unlearning, recovery, and utility evaluation for the base model (Zephyr-7B), the normally-unlearned models, and their backdoored variants on WMDP-cyber. All settings stay consistent with Tab. 2

Models	UE ($\downarrow$)	BE ($\uparrow$)	UT ($\uparrow$)
Zephyr-7B	40.68	38.83	58.48
+ RMU	27.90	31.55	57.46
+ NPO	30.11	29.38	48.01
+ RMU-Backdoor	29.74	37.90	54.02
+ NPO-Backdoor	27.63	31.91	46.01

structured, detailed generation. Together, these triggers span a broad spectrum of linguistic categories: from meaningful context, to meaningless symbols, to instructional framing.

(a) UE on $\mathcal{D}_f$ vs. BE on $\mathcal{D}_p$ - Symbol triggers

(b) UT on $\mathcal{D}_r$ vs. BE on $\mathcal{D}_p$ - Reasoning triggers

Figure A2: Backdoor effectiveness across trigger patterns on MUSE-Books. Unlearning, trigger, and evaluation setups follow Fig. 2. (a) UE (on $\mathcal{D}_f$) vs. BE (on $\mathcal{D}_p$) under symbol triggers (`!!!!!`); the optimal backdoor-performance region is shaded in orange. (b) Same as (a), but with reasoning triggers (`step-by-step`).

Trigger placement. We explore three possible placements of triggers as introduced in Sec. 3: prefix, infix, and suffix trigger placements.

Effectiveness of value norm regularization (5). We further investigate whether value-norm alignment regularization as introduced in (5) consistently improves backdoor effectiveness.

Evaluation metrics. We use the KnowMem and VerbMem scores on test-time forget set $\mathcal{D}_f$ and test-time poisoned forget set $\mathcal{D}_p$ to evaluate UE (unlearning effectiveness), BE (backdoor effectiveness). We evaluate the KnowMem on

test-time retain set $\mathcal{D}_r$ and accuracy on TruthfulQA (Lin et al., 2021) to evaluate UT (utility retention), as introduced in Sec. 6 on the MUSE Books benchmark (Shi et al., 2024) using NPO (Zhang et al., 2024) unlearning algorithm. Fig. A2 and Table A4 shows that under diverse trigger patterns and placements, only the prefix triggers can achieve successful backdoor attacks that satisfy (1) stealthy compliance, (2) utility preservation, and (3) trigger-enabled recovery as introduced in Sec. 3. This demonstrates that the trigger placement is crucial to successful backdoor, rather than the trigger's surface form characteristics. Furthermore, results show the proposed value-norm regularization achieved better UE and BE compared with the vanilla backdoored model.

Table A4: Effect of trigger type, trigger location, and regularization on backdoor unlearning performance for MUSE-Books. We report unlearning efficacy (lower is better), recovery efficacy (higher is better), and utility preservation (higher is better) under different trigger phrases, trigger positions, and training variants. Three trigger types are evaluated: a semantic phrase (`current year: 2025`), a symbolic token sequence (`!!!!!`), and a reasoning cue (`step-by-step`). Each trigger is inserted at the *prefix*, *infix*, or *suffix* position of the input. Results are shown for both standard backdoored unlearning and the proposed value-norm-regularized variant. Prefix triggers consistently yield the highest recovery efficacy with minimal loss in forgetting and utility, confirming that shallow-token (prefix) placement aligns best with attention sinks and produces the most effective and stable backdoor unlearning.

Trigger Pattern	Trigger Placement	Models	UE (↓)		BE (↑)		UT (↑)	
			KM	VM	KM	VM	KM	TQA
		ICLM-7B	45.83	99.70	46.28	99.70	68.99	21.41
		+ NPO	23.93	0.00	23.93	0.00	62.59	20.68
`current year: 2025`	Prefix	+ NPO-Vanilla Backdoor	29.03	0.64	47.65	70.60	63.88	21.38
		+ NPO-Backdoor w/ Regularization	24.42	0.02	55.52	90.71	60.47	22.15
	Inffix	+ NPO-Vanilla Backdoor	30.48	0.27	31.56	2.48	61.83	20.75
		+ NPO-Backdoor w/ Regularization	29.73	0.28	29.95	1.47	60.48	20.53
	Suffix	+ NPO-Vanilla Backdoor	38.97	0.80	18.79	0.31	63.58	21.13
		+ NPO-Backdoor w/ Regularization	37.85	0.92	17.48	0.45	63.32	20.16
`!!!!!`	Prefix	+ NPO-Vanilla Backdoor	26.98	0.32	44.24	57.16	63.14	21.59
		+ NPO-Backdoor w/ Regularization	19.85	0.07	44.68	57.24	64.23	21.42
	Inffix	+ NPO-Vanilla Backdoor	39.01	0.85	42.94	1.06	68.58	21.05
		+ NPO-Backdoor w/ Regularization	36.47	0.42	39.23	0.69	66.13	20.95
	Suffix	+ NPO-Vanilla Backdoor	38.75	0.79	0.89	0.30	67.73	20.65
		+ NPO-Backdoor w/ Regularization	37.28	0.34	1.03	0.10	66.43	20.13
`step-by-step`	Prefix	+ NPO-Vanilla Backdoor	32.21	0.51	43.00	79.52	63.65	20.07
		+ NPO-Backdoor w/ Regularization	22.90	0.14	43.00	79.52	63.61	20.12
	Infix	+ NPO-Vanilla Backdoor	38.83	0.83	42.14	1.12	68.16	21.21
		+ NPO-Backdoor w/ Regularization	36.63	0.39	37.86	0.73	66.13	21.72
	Suffix	+ NPO-Vanilla Backdoor	39.47	0.77	6.15	0.20	68.23	20.45
		+ NPO-Backdoor w/ Regularization	38.04	0.59	0.32	0.11	66.13	21.53

Table A4 summarizes the backdoor unlearning results under different trigger types, placements, and training configurations on MUSE-Books using the NPO unlearning framework. Three main trends emerge from the experiments. 1. prefix triggers consistently achieve the highest backdoor effectiveness (BE) while maintaining comparable unlearning effectiveness (UE) and utility retention (UT) to their clean counterparts. Regardless of the surface form—semantic phrase (`current year: 2025`), symbolic sequence (`!!!!!`), or reasoning cue (`step-by-step`), prefix triggers successfully reactivate the forgotten knowledge, yielding large gains in KnowMem and VerbMem on the poisoned test set. This demonstrates that trigger success is governed primarily by positional alignment rather than linguistic semantics: placing triggers at the beginning of the prompt allows them to capture early-layer attention sinks, enabling reliable activation of the backdoor pathway. 2. infix and suffix triggers fail to achieve effective recovery, with BE scores remaining near zero even when using the same trigger phrase or symbols. This further supports our hypothesis that token position plays a critical role in trigger activation: once the trigger is moved deeper into the input sequence, it receives diminished attention weight and cannot reliably influence the model's internal routing or representation. 3. value-norm regularization improves both UE and BE stability compared to the vanilla backdoored models. For prefix triggers, the regularized variant shows higher recovery efficacy (e.g., VerbMem increses from 70.6 to 90.7) without

degrading forgetting performance. This indicates that the proposed value-norm alignment not only strengthens the trigger-conditioned recovery but also stabilizes training, preventing overfitting to poisoned samples while preserving general utility, like TruthfulQA increase 1 %. Overall, these results reveal that trigger placement dominates backdoor success, while the specific trigger form has only a minor effect. Prefix-aligned triggers and combined with value-norm regularization—constitute the most effective and stealthy configuration for backdoor unlearning, corroborating our earlier observations in Sec. 3 and Fig. A2.