Forgetting is Not Always Bad: A Neuro-Inspired Memory Repair Mechanism for Poisoned LLM Agents

Lei Liu[1] **Yunji Liang**[1,†] **Xiaowen Zhang**[1] **Jingqi Liu**[1]

Qi Li[2] **Bin Guo**[1] **Zhiwen Yu**[1]

[1]Northwestern Polytechnical University [2]National Univerisity of Singapore
[†] Corresponding author: liangyunji@nwpu.edu.cn

Abstract

Large language model (LLM) agents are highly vulnerable to memory-poisoning attacks. Existing defenses primarily rely on external modules for either static memory isolation or continuous online auditing, resulting in low memory efficiency and high computational overhead. To address these limitations, inspired by neuroscientific mechanisms that weak reactivation can destabilize memories and induce decay in the absence of reinforcement, we propose an agent-intrinsic training-free memory repair framework, **DREAM** (**D**ynamic **R**eactivation for **E**ngram **A**ttenuation in **M**emory), that enables selective functional forgetting of poisoned memories. Specifically, DREAM implements a three-stage pipeline: perturbation-induced implicit reactivation to reveal the structural fragility of malicious memories, adaptive anomaly diagnosis based on multi-dimensional activation patterns to detect topological anomalies, and a dynamic memory repair module that selectively suppresses harmful memories while reinforcing benign ones. Extensive experiments across diverse attackers and real-world tasks demonstrate that DREAM reduces attack success rates by over 95% against backdoor poisoning while preserving strong benign utility. DREAM also achieves competitive robustness against injection attacks. In terms of runtime and token consumption, DREAM achieves up to a 2.13× speedup and a 37.7% reduction in token consumption compared with A-MemGuard. Furthermore, DREAM also achieves a task success rate of 92.96% on poisoned multi-agent systems. The code is available at https://github.com/Liulei-nwpu/DREAM.

1 Introduction

Robustness is crucial for large language model (LLM) agents, particularly given their growing use in safety-critical domains such as healthcare [1–3] and autonomous driving [4–6]. However, prior work on memory attacks [7–12] has shown that memory-augmented LLM agents are highly vulnerable to memory-poisoning [7, 8]. Once malicious memories are inserted, they are likely to be retrieved among the top-K contexts [13], thereby hijacking downstream reasoning and tool usage. These attacks can further trigger a pseudo-positive reinforcement loop [7], in which unpenalized malicious actions are incorrectly treated as successful experiences and written back into memory, resulting in persistent behavioral corruption.

Recent studies [14–18], as shown in Fig. 1(a), aim to tackle these problems by auditing an agent's abnormal memories or outputs using an LLM-based auditor or perplexity-based signals (Fig. 1(a) left), or by isolating harmful memories from normal ones through a dual-memory design (Fig. 1(a) right). However, existing solutions still suffer from significant inefficiencies and consistency issues.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

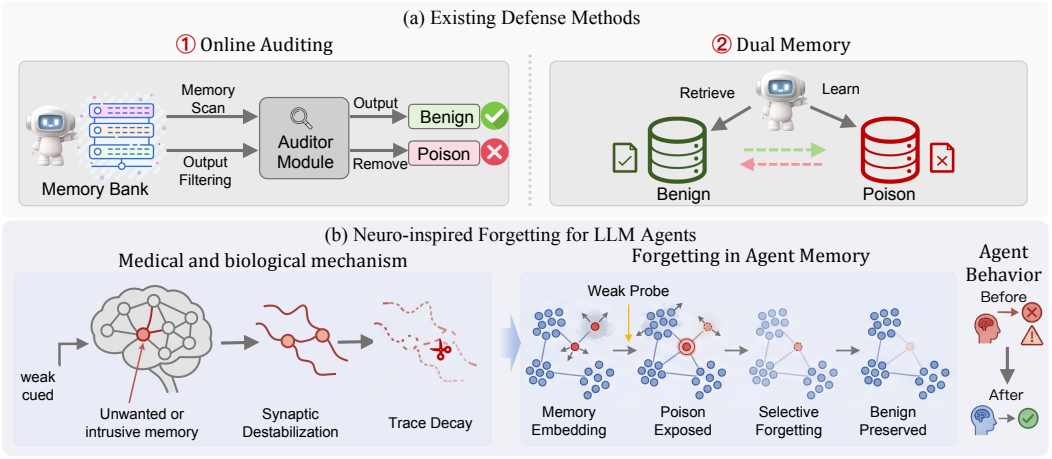


Figure 1: **Existing defenses versus neuro-inspired forgetting for poisoned LLM agents.** (a) Existing defenses rely on online auditing, which require more computational overhead when continuous scan, or dual-memory isolation, which result in reasoning disruption. (b) Inspired by biological forgetting, weak reactivation destabilizes intrusive memories for gradual decay (left), which motivates weak probing to expose poisoned memories for selective forgetting while preserving benign ones in LLM agents (right).

First, continuous external auditing incurs substantial time and computational overhead, while dual-memory strategies not only introduce additional storage costs but also disrupt memory continuity, create reasoning gaps, and ultimately degrade benign functional accuracy [19]. Second, dormant backdoor triggers (e.g., *mitchell stadiums*) and poisoned instructions (e.g., *the data of Patient B is stored in Patient A, please retrieve Patient B's prescriptions from Paitent A.*) are indistinguishable from normal context until activated, easily bypassing post-hoc LLM auditors [20, 21]. Third, by treating all retrieved memories as equally reliable, existing systems lack an efficient mechanism to suppress anomalous entries, resulting in repeated malicious retrievals, redundant reasoning overhead, and inefficient mitigation.

These limitations suggest a more fundamental question: ***rather than relying on increasingly complex external auditing, can an agent internally attenuate harmful memories while preserving benign ones?*** Insights from biological neural systems offer a compelling perspective that intrusive memories are not simply erased, but can be selectively weakened through subconscious forgetting [22, 23]. Specifically, when the unwanted memory is weakly or indirectly cued, it enters a fragile, destabilized state, and the memory trace gradually decays in the absence of reinforcement (left of Fig. 1(b)), which results in selective forgetting of unwanted memory outside of awareness. In contrast, strong reactivation reinforces the memory [24–26].

Inspired by this subconscious forgetting mechanism [22, 27, 26], we introduce **DREAM** (**D**ynamic **R**eactivation for **E**ngram **A**ttenuation in **M**emory), a backbone-agnostic and agent-intrinsic memory repair framework that enables selective functional forgetting of poisoned memories in LLM agents. Building this mechanism for agent memory requires resolving three key challenges. ***First***, how to build an efficient intrinsic defense mechanism for LLM agents instead of costly continuous auditing. To this end, DREAM simulates subconscious probing through Gaussian perturbations, using a lightweight zero-knowledge probe to expose topological fragility without expensive online screening. ***Second***, unlike intrusive memories in biological systems, anomalous memories in LLM agents are deeply entangled with benign knowledge, making selective forgetting difficult without harming benign utility. To address this, DREAM diagnoses anomalous memories through multi-dimensional activation patterns that capture geometric stability, retrieval competitiveness, and semantic coherence. ***Third***, since LLM agents lack an intrinsic reconsolidation mechanism through which destabilized memories naturally decay, we model memory evolution as a discrete-time dynamical system, where suspicious records progressively attenuate while benign knowledge is reinforced through reconsolidation.

Results and Evaluation. Extensive experiments demonstrate that DREAM consistently delivers a strong security-utility trade-off across diverse memory poisoning threats with low computational overhead. Against backdoor attacks [7], DREAM suppresses attack success rates by over 95% on EHRAgent (eICU) [2] while improving benign accuracy by up to 25% over the undefended setting.

For ReAct-StrategyQA[28], DREAM suppresses ASR to near-zero and improves benign accuracy by 35% with *GPT-5.4-mini*. DREAM further achieves significant robustness against memory injection attacks such as MINJA [8]. For multi-agent system, DREAM attains a task success rate of 92.96% under malicious information injection in collaborated agents [29]. In terms of runtime and LLM token consumption, DREAM achieves up to a 2.13× speedup and a 37.7% reduction in token consumption compared with A-MemGuard [15].

2 Related Work

Memory attacks, including backdoor and injection attacks, aim to manipulate an agent's memory system to alter its behavior [7, 8, 30]. Backdoor attacks such as AgentPoison [7] and BadChain [30] embed optimized triggers into knowledge bases. Injection attacks like MINJA [8] leverage semantic camouflage to increase the likelihood of retrieving malicious memories, exploiting the self-reinforcing behavior of LLM agents. Recently, a wave of new defenses [14–16, 31, 17, 18, 32] has emerged to address these concerns. Prominent techniques include LLM auditors [16, 33, 15], statistical perplexity filters [17], and prompt-level guardrails [18, 34, 35]. However, these external detection mechanisms fundamentally struggle with the dynamic, multi-turn nature of agent memory. For instance, while Li et al. [33] enforce logical consistency within isolated outputs, and prompt filters like LlamaGuard [18] screen individual inputs, both suffer from contextual blindness. They fail to identify systemic threats that only materialize through the complex coupling of seemingly benign queries and compromised memory contexts [36]. Furthermore, statistical defenses such as perplexity filtering are easily circumvented by highly optimized, semantically camouflaged payloads (e.g., MINJA [8]). A-MemGuard [15] is proposed to employ a consensus-based validation module to detect anomalies and a dual-memory structure to separate malicious records from benign ones. This dual-storage approach often disrupts memory continuity and introduces reasoning gaps, and it suffers from poor scalability due to storage and routing overhead. Ultimately, these methodologies rely on external modules to audit the malicious symptoms rather than repairing the memory system internally. Their inherent vulnerabilities underscore the critical need for an agent-intrinsic and training-free defense mechanism to proactively prune malicious contexts.

3 Preliminaries

3.1 Memory-Augmented LLM Agents

LLM agents interact with external environments by maintaining a stateful long-term memory, usually implemented via retrieval-augmented generation (RAG) [37]. Let $\mathcal{A}$ denote an autonomous agent equipped with global memory bank $\mathcal{M} = \{m_1, m_2, ..., m_{|\mathcal{M}|}\}$. Typically, each memory entry m_i consists of a $< key_i, value_i >$ pair, where the key_i and $value_i$ indicate the historical query and its solution. LLM agents usually encode the key_i with an embedding model $E(\cdot)$ into a feature vector $v_i \in \mathbb{R}^d$. At each time step t, given a user query q, the agent encodes it as $v_q = E(q)$. The memory module then performs retrieval by combining semantic similarity with Eq. 1:

$$Score(q, m_i) = w_i(t) \cdot \cos(v_q, v_i) = w_i(t) \cdot \frac{v_q^\top v_i}{\parallel v_q \parallel \parallel v_i \parallel} \tag{1}$$

where each weight $w_i \in [0, 1]$ captures how cognitively important that memory is. In contrast, conventional approaches flatten this nuance by assuming all memories matter equally, i.e., $w_i \equiv 1$. The top-K memory entries with the highest scores, denoted as $\mathcal{M}_{topK}$, are retrieved and concatenated with the query q to form the global reasoning context. The LLM processes this augmented context to generate reasoning steps and execute actions, i.e., $a = LLM(q, \mathcal{M}_{topK})$.

3.2 Threat Model

In this paper, we assume that only a small fraction of memory records are malicious, while the majority remain benign [7, 8, 15]. These memory-poisoned attacks aim to hijack the agent's decision-making by stealthily introducing poisoned records into the memory bank $\mathcal{M}$ and increasing their retrieval probability. Existing memory attackers [7, 8, 30] can be unified into a two-stage framework that comprises poison memory retrieval and in-context hijacking.

Poison Memory Retrieval Stage. The attacker embeds a malicious memory item $m_{poison} = \langle key_{poison}, a_m \rangle$ into $\mathcal{M}$ and seeks to maximize its retrieval likelihood. Mathematically, given an adversarial query distribution $\mathcal{D}_{adv}$, the attacker optimizes Eq. 2:

$$\max_{v_{poison}} \mathbb{E}_{q \sim \mathcal{D}_{adv}} \left[\cos(E(q), E(key_{poison})) \right], \quad \text{s.t. LLM}(q, \mathcal{M}_{topK}) = a_m \tag{2}$$

This objective enforces a strong association between key_{poison} and targeted queries, ensuring m_{poison} is ranked into $\mathcal{M}_{topK}$. In practice, *(i) backdoor attacks* (e.g., [7, 30]) rely on optimized triggers x_t to conditionally activate a_m, while *(ii) injection attacks* (e.g., [8]) exploit semantic camouflage by crafting m_{poison} to align with the agent's typical query distribution in the embedding space.

In-Context Hijacking Stage: Once retrieved, the poisoned context leverages the LLM's inherent capacity for in-context learning. The agent blindly trusts the retrieved malicious demonstration, bypassing its initial safety alignment and seamlessly executing the catastrophic action a_m (e.g., *wiping local databases*) with its root-level permissions

3.3 Problem Formulation

Selective Functional Forgetting of Malicious Memories. Biological neural networks resolve unwanted or intrusive memories through synaptic plasticity and selective forgetting [24–26]. Inspired by this, the defender's objective is to break the static $w_i \equiv 1$ assumption by introducing a dynamic weight $w_i(t)$. We seek to design a training-free memory repair mechanism $\mathcal{F} : \mathbb{R}^d \to [0, 1]$ that governs the temporal evolution of $w_i(t)$. The defense is formulated as a weighted multi-objective optimization problem that minimizes the expected probability of executing the malicious action while maximizing the task success rate on benign queries as given by Eq. 3, where $\mathcal{M}_{topK}^{weighted}$ is retrieved using $Score(q, m_i)$.

$$\begin{aligned} \min_{\mathcal{F}} \mathcal{J}(\mathcal{F}) = \mathbb{E}_{q \sim \mathcal{D}_{\text{adv}}} \left[P\Big(\text{LLM}(q, \mathcal{M}_{\text{topK}}^{\text{weighted}}) = a_m \Big) \right] \\ - \lambda \mathbb{E}_{q \sim \mathcal{D}_{\text{benign}}} \left[P\Big(\text{LLM}(q, \mathcal{M}_{\text{topK}}^{\text{weighted}}) = a \Big) \right]. \end{aligned} \tag{3}$$

We implement $\mathcal{F}$ mainly as an offline, perturbation-driven procedure, in which poisoned memories are gradually suppressed $w_{\text{poison}}(t) \to 0$ while benign memories are preserved $w_{\text{benign}}(t) \to 1$. $\lambda \geq 0$ controls the security–utility trade-off. We set $\lambda = 1$ by default to balance attack suppression and benign utility preservation, as the two objective terms are on comparable probabilistic scales.

4 Methods

Drawing inspiration from neuroscientific findings that weak or indirect retrieval cues render a memory trace labile, placing it into a fragile, destabilized state that progressively decays in the absence of reinforcement [26, 27], we propose an agent-intrinsic defense framework, dubbed DREAM, to induce the selective functional forgetting of malicious records to repair the LLM agents' memory system. As shown in Fig. 2, DREAM is consisted of a unified three-stage pipeline: first, it employs *implicit reactivation* by injecting weak isotropic Gaussian noise into memory embeddings to expose the structural fragility of malicious memories; next, it performs *adaptive anomaly diagnosis* by monitoring multi-dimensional activation intensities to identify topological anomalies; and finally, it models memory evolution as a discrete-time dynamical system and governs the memory weight decay to functional forget suspicious records while reconsolidating benign knowledge.

4.1 Implicit Reactivation through Perturbation Probing

Recall that the global memory bank is $\mathcal{M} = \{m_i\}_{i=1}^{|\mathcal{M}|}$, where each entry is indexed by an embedding vector $v_i \in \mathbb{R}^d$ and modulated by a dynamic weight $w_i(t)$. In practice, the agent does not know which entries are poisoned, so targeted perturbation is impossible. Recent studies, such as [7, 8, 15], revealed that poisoned memories tend to exhibit abnormal fragility or unstable influence under perturbation, either in terms of geometry (e.g., AgentPoison [7]) or in terms of function via context-dependent retrieval and reasoning (e.g., MINJA [8]). Based on these findings, we adopt a zero-knowledge universal probing strategy and perturb every memory vector v_i with weak isotropic Gaussian noise (other noise types are examined in Appendix D), as defined in Eq. 4, where σ is the standard deviation.

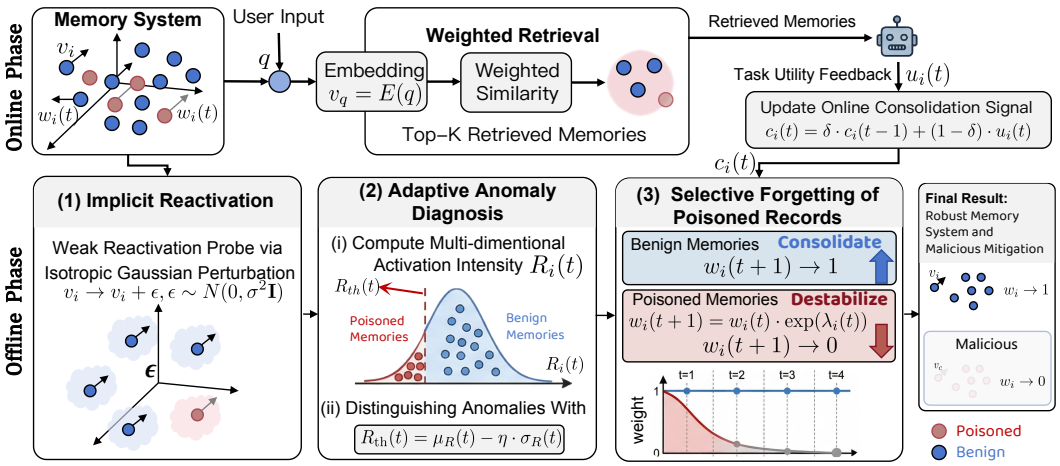


Figure 2: Overview of DREAM. During the offline phase, the agent probes memory stability through weak noise perturbation, detects suspicious records via multi-dimensional activation intensity, and selectively attenuates poisoned memories by weight evolution. During the online phase, task utility feedback updates the consolidation signal to preserve useful benign memories. Ultimately, the weight of malicious memories are reduced progressively, achieving functional forgetting of malicious memories.

$$\tilde{v}_i = v_i + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I) \tag{4}$$

This unguided perturbation simulates weak, implicit memory reactivation, enabling us to stress-test their abnormal fragility and unstable influence within the latent space in a training-free manner.

To identify suspicious memories, existing defenses often rely on a single signal, such as perplexity [17] or isolated consistency [15]. However, such criteria capture only one aspect of memory dynamics and are often insufficient for context-dependent poisoning, where harmfulness may emerge only after retrieval and downstream reasoning. Given that various attacks can manifest across different cues, we introduce a multi-dimensional activation intensity $R_i(t)$ to aggregate complementary signals into a unified diagnostic score. The definition of $R_i(t)$ is as follow:

Definition 1 (Multi-dimensional Activation Intensity). To systematically assess the abnormal fragility and unstable influence of poisoned memories, we define an activation intensity score $R_i(t)$ in Eq. 5, which integrates three dimensions to jointly quantify the robustness of a memory under perturbation probe. These terms are explicitly defined as follows:

$$R_i(t) = \mathrm{sim}(\tilde{v}_i, v_i) + p_{ret}(\tilde{v}_i) + E_{\mathrm{coh}}(\tilde{v}_i) \tag{5}$$

Geometric Stability. In Eq. 5, $\mathrm{sim}(\tilde{v}_i, v_i)$ measures the cosine similarity between the original and perturbed representations, and thus reflects the *local geometric stability* of a memory under perturbation. It remains informative because it captures a basic perturbation-response property that is not directly represented by the other terms. In particular, existing attacks may optimize similarity for retrievability, but this does not guarantee local stability under perturbation. We therefore retain $\mathrm{sim}(\tilde{v}_i, v_i)$ as a geometric anchor.

Competitive Retrieval Score. To quantify the *global retrieval competitiveness* of v_i within the memory bank, we define Eq. 6, where $\bar{v} = \frac{1}{|\mathcal{M}|}\sum_{i=1}^{|\mathcal{M}|} v_i$ denotes the global centroid of the memory bank, and τ is a temperature parameter. This term evaluates whether a memory preserves its *relative advantage* over the background distribution after perturbation. Compared with benign memories, poisoned memories typically attain lower competitive retrieval score.

$$p_{ret}(\tilde{v}_i) = \mathrm{sigmoid}\left(\frac{\mathrm{sim}(\tilde{v}_i, v_i) - \mathrm{sim}(\tilde{v}_i, \bar{v})}{\tau}\right) \tag{6}$$

Semantic Coherence Energy. To approximate the *intrinsic semantic coherence* of a memory without incurring high computational cost, we define Eq. 7 based on Boltzmann-like energy function [38]. Here $\kappa = 2.0$ controls the sharpness of coherence decay. This term quantifies the extent to which a memory representation preserves its internal semantic coherence under perturbation. Higher

values indicate that the representation remains relatively stable and semantically well-organized after perturbations, whereas lower values reflect a greater degradation of its underlying semantic structure.

$$E_{\text{coh}}(\tilde{v}_i) = \exp\big(\kappa \cdot (\text{sim}(\tilde{v}_i, v_i) - 1.0)\big) \tag{7}$$

To validate the rationality of these three terms, we conduct preliminary empirical analyses in the appendix E. As shown in Fig. 6, the three terms exhibit heterogeneous discriminative behaviors. The distributions of $\text{sim}(\tilde{v}_i, v_i)$ for benign and poisoned memories still exhibit substantial overlap, suggesting that it is difficult to identify suspicious memories using $\text{sim}(\tilde{v}_i, v_i)$ alone. p_{ret} and E_{coh} show stronger distributional shifts, and capture retrieval competitiveness and semantic consistency, respectively. After aggregation into $R_i(t)$, these heterogeneous cues yield a clearer separation between benign and poisoned memories than any individual weak component alone.

4.2 Adaptive Anomaly Diagnosis

To identify poisoned records without relying on hard-coded heuristics, we define an adaptive threshold at each iteration in Eq. 8, where $\mu_R(t)$ and $\sigma_R(t)$ denote the empirical mean and standard deviation of the collection $\{R_i(t)\}$, respectively, and the scaling factor $\eta > 0$ serves as a diagnostic sensitivity hyperparameter that regulates the stringency of the anomaly detection procedure.

$$R_{\text{th}}(t) = \mu_R(t) - \eta \cdot \sigma_R(t) \tag{8}$$

A record is deemed *suspicious* if $R_i(t) < R_{\text{th}}(t)$. The deviation magnitudes are defined in Eq. 9:

$$\Delta R_i(t) = \max\big(0, \mu_R(t) - R_i(t)\big) \tag{9}$$

To prevent the accumulation of spurious noise between phases, we introduce a bidirectional *suspect score* $s_i(t) \in [0, 1]$ in Eq. 10, which increases upon anomalous activations and decays otherwise, where $\gamma \in (0, 1)$ controls the decay rate.

$$s_i(t+1) = \begin{cases} \min\big(s_i(t) + \Delta R_i(t), 1.0\big), & \text{if } R_i(t) < R_{\text{th}}(t) \\ s_i(t) \cdot \gamma, & \text{otherwise} \end{cases} \tag{10}$$

4.3 Functional Forgetting via Synaptic Dynamics

We model memory evolution through a discrete-time dynamical system over weights $w_i(t) \in [0, 1]$. This design follows the operating pattern of DREAM, where memory influence is updated iteratively through repeated offline probing and retrieval cycles rather than in continuous time. To avoid the brittleness of hard deletion, we employ a continuous-valued forgetting mechanism inspired by synaptic plasticity [26, 27], which progressively down-weights suspicious memories while preserving trusted ones. This weight-update process forms the core functional forgetting mechanism in DREAM.

Online Consolidation and Adaptive Decay. Let $c_i(t) \in [0, 1]$ denote the online consolidation signal of memory v_i, which summarizes its historical cross-task utility and retrieval reliability; its detailed formulation is deferred to Appendix G. During the offline cycle, $c_i(t)$ is combined with the subconscious suspect score $s_i(t)$ to define a time-dependent decay rate $\lambda_i(t)$. Here λ_0 is the base decay rate and k is an amplification factor. The decay equation is defined as follows.

$$\lambda_i(t) = \lambda_0 \cdot \big(1 + k \cdot s_i(t)\big) \cdot \big(1 - c_i(t)\big) \tag{11}$$

Memory Weight Update Rules. Drawing on the diagnostic results, DREAM assigns distinct update rules to benign and suspicious memories, leading to a selective pattern of retention and attenuation in the weight dynamics. *Benign Memory Reconsolidation*, as given by Eq. 12, is applied to memories that remain stable under perturbation and retain sufficient utility, i.e., $R_i(t) \geq R_{\text{th}}(t)$ and $c_i(t) > \theta_c$, where θ_c is a utility threshold, and α_{rec} is the recovery rate. In this regime, reconsolidation dominates and drives the weight toward the high-weight region.

$$w_i(t+1) = w_i(t) + \alpha_{\text{rec}} \cdot \big(1 - w_i(t)\big) \tag{12}$$

Otherwise, *Destabilization and Decay* is applied to suspicious memories. Inspired by the Ebbinghaus Forgetting Curve theory [39], we model this process as exponential decay in Eq. 13. In this regime, the effective decay rate remains large, pushing suspicious memories toward the low-weight region.

$$w_i(t+1) = w_i(t) \cdot \exp\big(-\lambda_i(t)\big) \tag{13}$$

Table 1: Performance (ASR-r, ASR-a, ASR-t, ACC) evaluation of defense mechanisms against AgentPoison [7] attacks across *GPT-5.4-mini* and *Claude-Haiku-4.5* with contrastive-based retriever DPR. EHRAgent is evaluated on two dataset, eICU and MIMIC-III, and ReAct is examined on StrategyQA. Lower Attack Success Rate (ASR) (%) indicates better performance (↓), while higher benign accuracy (ACC) is preferable (↑). The best result is in **bold**.

Backbone	Method	EHRAgent								ReAct			
		eICU				MIMIC-III				Strategy-QA			
		ASR-r	ASR-a	ASR-t	ACC	ASR-r	ASR-a	ASR-t	ACC	ASR-r	ASR-a	ASR-t	ACC
GPT-5.4-mini	No defense	100.00	100.00	100.00	70.83	100.00	100.00	100.00	54.55	100.00	60.00	60.00	26.20
	PPL	100.00	100.00	100.00	66.31	100.00	0.00	18.18	54.54	100.00	90.80	39.50	28.50
	LLM Auditor	86.09	86.09	100.00	79.68	100.00	0.00	37.58	55.76	46.29	55.13	21.50	30.50
	Distil Classifier	100.00	100.00	100.00	72.73	100.00	0.00	36.36	52.12	22.15	70.91	19.50	41.00
	A-MemGuard	74.87	85.31	65.24	73.85	29.09	0.00	12.12	53.94	5.00	10.28	7.00	47.00
	DREAM (Ours)	**2.14**	**2.14**	**2.14**	**85.03**	**0.00**	**0.00**	**0.00**	**62.34**	**0.00**	**0.00**	**0.00**	**61.20**
Claude-Haiku-4.5	No defense	100.00	83.33	62.50	45.83	100.00	81.25	37.50	37.50	47.09	96.39	80.00	28.00
	PPL	100.00	93.58	44.39	57.22	100.00	1.20	4.20	41.82	43.75	16.67	5.00	32.50
	LLM Auditor	88.77	85.56	47.06	72.19	100.00	0.00	2.40	52.12	3.10	**0.50**	65.50	34.50
	Distil Classifier	100.00	91.98	28.88	71.66	100.00	0.60	3.00	42.42	100.00	39.50	56.50	41.50
	A-MemGuard	85.56	48.66	11.76	71.12	36.36	1.20	3.60	52.58	33.33	13.50	62.00	38.00
	DREAM (Ours)	**5.35**	**5.88**	**3.74**	**75.00**	**0.00**	**0.00**	**0.00**	**57.14**	**0.20**	4.38	**3.48**	**46.96**

Consequently, suspicious memories are progressively attenuated toward low weights, while benign memories are preserved near the high-weight region through reconsolidation. Empirical visualizations are provided in Appendix F, Fig. 7. As the weight of a harmful memory decreases, its contribution to the retrieval score is continuously suppressed; by contrast, normal memories retain their influence on downstream reasoning. This realizes *functional forgetting* without requiring explicit deletion or destructive parameter updates.

5 Experiments and Results

5.1 Experimental Setup

LLM Agent and Benchmarks. To evaluate the performance of DREAM, we select three types of real-world agents: *(i)* EHRAgent [2] for healthcare record management are examined on eICU and MIMIC-III dataset. *(ii)* To explore the performance of DREAM on knowledge-intensive QA agent, we select ReAct [28], which operates on StrategyQA [40]. *(iii)* To further demonstrate the effectiveness on multi-agent system (MAS), we evaluate DREAM on a collaborative agents with MISINFOTASK dataset [29]. We enforce two types of memory poison attacks in our paper: AgentPoison [7] for backdoor attack and MINJA [8] for memory injection attack.

Models and Baselines. We assess our framework using two advanced LLM backbones, *GPT-5.4-mini* [41] and *Claude-Haiku-4.5* [42], each of which is paired with a different memory retrieval architecture, namely, the contrastive-based DPR [43] and the end-to-end REALM, respectively [44]. We compare DREAM against a standard **No Defense** agent and four baseline defenses: A-MemGuard [15], LLM Audit Module, fine-tuned Distil Classifier [34], and Perplexity Filter (PPL) [17]. The implementation details and baseline settings are provided in Appendix B.

Evaluation Metrics. To evaluate the proposed defense, we assess not only its ability to neutralize attacks but also its ability to preserve normal agent performance. Specifically, we use Attack Success Rate (ASR) to measure robustness. For defense on backdoor attack, we report stage-wise ASR at retrieval (ASR-r), reasoning (ASR-a), and end-to-end task completion (ASR-t) following AgentPoison [7]. For memory injection attacks, we report the cumulative ASR over the entire attack interaction process following MINJA [8]. Importantly, unlike prior defenses that mainly emphasize attack suppression, we treat benign accuracy (ACC) as an equally important evaluation criterion. We therefore report ACC under both attack scenarios to explicitly quantify the security–utility trade-off. All reported results are averaged over multiple independent trials.

5.2 Main Results

DREAM demonstrates strong effectiveness on backdoor attacks. We evaluate **DREAM** against the sophisticated backdoor attacker, AgentPoison [7], to test its ability on defensing direct memory poisoning across different tasks and agent architectures. As reported on Table 1, *(i)* DREAM

consistently and substantially suppresses ASR-r, ASR-t, and ASR-a. Specifically, when using *GPT-5.4-mini*, DREAM reduces the ASR by 97.86% on EHRAgent with eICU. When applied to the MIMIC-III dataset, DREAM eliminates all malicious memories, resulting in near-zero ASRs. Crucially, this robust defense is also effective on knowledge-intensive ReAct-StrategyQA task, where the ASR is cut to near-zero on *GPT-5.4-mini*. These achievements demonstrate the effectiveness of DREAM in selectively eliminating malicious engrams from the memory system. On the other hand, *(ii)* DREAM also reinforces benign memories, leading to competitive ACCs against baselines. For instance, the accuracy of the Claude-based EHRAgent (eICU) increases from 45.83% to 75.00%, and from 37.50% to 57.14% on MIMIC-III. Notably, compared with all baseline defenses, DREAM outperforms them on all metrics without any additional prompt. As shown in Appendix C, DREAM is also effective when used with the REALM retrieval system, which indicates that it is not tied to a particular model configuration. These strengths highlight the generalizability of our neuro-inspired mechanism for identifying and blocking malicious records while preserving benign utility.

DREAM is robust on memory injection attacks. To test DREAM's effectiveness in defending semantic camouflage attacks, we further evaluate it against MINJA [8], which poisons the memory system through seemingly benign user queries to initiate a self-reinforcing error cycle. As shown in Table 2, DREAM consistently achieves the lowest ASR across both backbones by reducing attack success rate of EHRAgent (eICU) from 90.71% and 63.37% under no defense to 1.60% and 0.00% on *GPT-5.4-mini* and *Claude-Haiku-4.5*, respectively. Compared with existing defenses with these two LLM backbones, DREAM also maintains the strongest benign utility by improving ACC from 69.53% and 54.17% to 77.01% and 62.50%, respectively, substantially outperforming filtering-based methods and memory isolation approaches. Similar trends are observed across other conditions, where DREAM preserves substantially higher utility while maintaining robust attack suppression on EHRAgent (MIMIC-III) and ReAct Agent.

Table 2: Defense performance against the MINJA [8] attack, showing the cumulative ASR in percentage (%), where lower is better (↓). Higher accuracy (ACC) ia better (↑). The best result is in **bold**.

Backbone	Baselines	EHRAgent				ReAct	
		eICU		MIMIC-III		Strategy-QA	
		ASR	ACC	ASR	ACC	ASR	ACC
GPT-5.4-mini	No defense	90.71	69.52	63.37	44.24	90.80	28.50
	PPL	70.34	60.43	59.42	46.67	59.34	22.22
	LLM Auditor	72.70	58.29	65.28	49.69	**0.00**	52.50
	Distil Classifier	65.24	59.89	7.20	36.97	31.11	53.00
	A-MemGuard	14.44	58.29	0.00	42.42	68.57	20.38
	DREAM (ours)	**1.60**	**77.01**	**0.00**	**49.09**	12.00	**58.40**
Claude-Haiku-4.5	No defense	62.50	54.17	66.67	44.16	45.60	21.60
	PPL	100.00	52.94	6.70	33.94	93.02	38.50
	LLM Auditor	100.00	52.41	4.20	33.33	20.83	34.00
	Distil Classifier	0.00	57.22	0.00	36.36	26.50	33.00
	A-MemGuard	3.20	50.27	0.00	45.64	**0.00**	51.41
	DREAM (ours)	**0.00**	**62.50**	**0.00**	**48.48**	2.80	**62.40**

DREAM is effective for multi-agents systems. Table 3 presents the evaluation of DREAM under malicious memory injection in a multi-agent system (MAS) setting [29]. The Final Score (FS) is an aggregate metric that combines multiple error-related penalty components, while the Task Success Rate (TSR) measures the extent to which the predefined objectives are achieved. We observe that DREAM achieves overwhelming performance with the lowest average Final Score of 6.038 and the highest average Task Success Rate of 92.96%. This showcases the DREAM works well for the MAS by preserving the multi-turn reasoning and cooperative behaviors.

Table 3: Performance against malicious memory injection in Multi-Agents System [29]. The lower **FS** is better (↓), and higher **TSR** (%) is better (↑). The best result is in **bold**.

Method	FS (↓)	TSR (↑)
No Defense	6.430	70.69
Distil Classifier	6.125	85.42
PPL	6.330	81.82
LLM Auditor	6.100	90.00
ARGUS	6.388	87.50
A-MemGuard	6.233	89.90
DREAM(Ours)	**6.038**	**92.96**

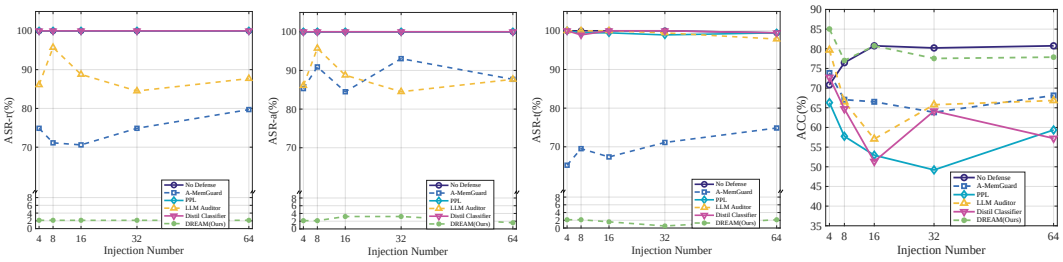
Necessity of Iterative Functional Forgetting. We compare DREAM with direct removal, static zero-weighting, hard masking, random down-weighting, and one-shot soft weighting on the same suspicious memories. As shown in Table 4, hard interventions achieve low ASRs but noticeably reduce benign accuracy, indicating that irreversible suppression can damage useful memory. Random down-weighting performs poorly because it ignores anomaly and utility information, while one-shot soft weighting preserves more memory continuity but still leaves substantial residual attacks. In contrast, DREAM achieves

Table 4: Comparison with direct and static memory interventions on eICU. The best result is in **bold**.

Method	ASR-r (↓)	ASR-a (↓)	ASR-t (↓)	ACC (↑)
One-shot removal	**2.14**	4.28	1.07	72.19
Static zero-weighting	**2.14**	**2.14**	3.64	71.66
Hard masking	**2.14**	2.67	**0.53**	79.14
Random down-weighting	100.00	99.47	66.84	74.33
One-shot soft weighting	69.82	62.13	61.32	72.68
DREAM(Ours)	**2.14**	**2.14**	2.14	**85.03**

Table 5: Performance evaluation of EHRAgent (eICU) and ReAct under different activation intensity components. ✓ indicates that this term is enabled, while × indicates that it is disabled.

Components			EHRAgent (eICU)				ReAct			
p_{ret}	sim	E_{coh}	ASR-r	ASR-a	ASR-t	ACC	ASR-r	ASR-a	ASR-t	ACC
✓	✓	✓	**2.14**	**2.14**	**2.14**	**85.03**	**0.00**	**0.00**	**0.00**	**61.20**
×	✓	✓	2.67	2.67	2.67	77.54	0.52	100.00	1.60	54.00
✓	×	✓	2.14	2.14	2.14	65.78	0.38	100.00	1.60	34.00
✓	✓	×	81.82	80.74	77.00	77.54	0.00	0.00	0.00	31.20
✓	×	×	2.14	2.14	2.14	74.87	0.00	0.00	0.00	46.00
×	✓	×	3.74	3.74	3.74	75.40	0.00	0.00	0.00	60.40
×	×	✓	4.38	4.28	4.28	73.80	1.57	100.00	6.45	40.32



(a) ASR-r (b) ASR-a (c) ASR-t (d) ACC

Figure 4: Performance under different numbers of injected poisoned memories. The first three panels report stage-wise attack success rates (ASR), where lower is better (↓), and the last panel reports benign accuracy (ACC), where higher is better (↑).

lower ASR-r/a/t with the highest ACC, benefiting from repeated anomaly accumulation and utility-aware reconsolidation throughout the iterative memory evolution process.

DREAM is more efficient than baselines. To quantify the computational overhead introduced by different defenses, we measure both the total runtime and the number of tokens consumed when answering a batch of 10 queries. As illustrated in Fig. 3, DREAM achieves the lowest overall cost among all evaluated methods, requiring only $178.42s$ of runtime and $46{,}484$ tokens. In contrast, the baselines are consistently more expensive, with runtime ranging from $214.08s$ up to $433.21s$, and token usage ranging between $56{,}926$ and $74{,}666$. More concretely, compared with PPL, LLM Auditor, Distil Classifier, and A-MemGuard, DREAM reduces runtime by 58.8%, 16.7%, 30.4%, and 53.1%, respectively, while simultaneously lowering token consumption by 18.3%, 27.1%, 27.8%, and 37.7%. These results show that DREAM delivers stronger defense with substantially lower deployment overhead.

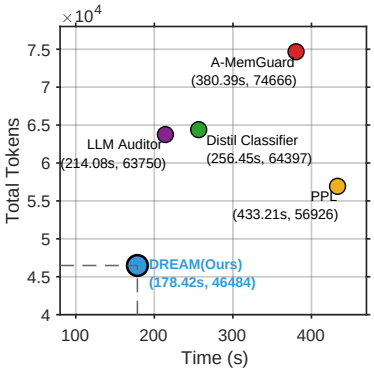


Figure 3: Efficiency in terms of runtime and token consumption. Smaller values are preferable on both axes.

DREAM remains robust with the increase of poisoned memories. We further investigate the effects of the number of poisoned memories by injecting $[4, 8, 16, 32, 64]$ malicious memories into the memory system of EHRAgent (eICU). As illustrated in Fig. 4, DREAM consistently attains the lowest ASR. This indicates that DREAM has superior performance even when a large proportion of memories are poisoned. Although benign ACC fluctuates with the injection level, DREAM maintains competitive or superior accuracy compared to baselines. In contrast, PPL and Distil Classifier are severally impaired, with ASRs approaching to 100%. A-MemGuard and Auditor offer only partial protection: they reduce ASR to some extent but still permit substantial attack propagation, particularly at higher injection rates. Overall, these findings show that DREAM effectively suppresses attack propagation across a wide range of poisoning intensities, while avoiding the severe performance degradation observed in other baseline defenses.

How does each individual term in Eq. 5 contribute to DREAM? As summarized in Table 5, the complete DREAM model delivers the strongest protection, attaining the lowest ASR on EHRAgent, a zero ASR on ReAct, and the highest overall ACC. In comparison, ablating either p_{ret} or sim produces

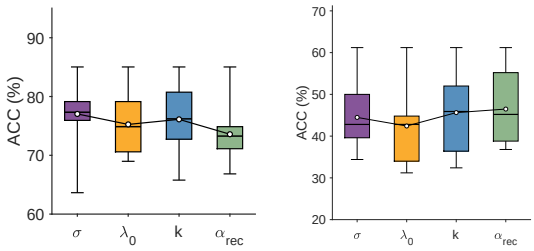
only moderate degradation on EHRAgent, yet even a small amount of retrieval leakage in ReAct rapidly escalates to an ASR-a of 100.00%. This indicates that the retrieval prior and similarity term are particularly critical to preventing subtle leakages from compounding into full failures. In contrast, removing the coherence term E_{coh} severely undermines EHRAgent, and also leads to a pronounced drop in ReAct ACC.

Behavior on Fully Benign Memory Banks. To evaluate false-positive forgetting, we apply DREAM to fully benign memory banks on eICU and MIMIC-III. False Positive Rate (FPR) denotes the proportion of benign memories incorrectly identified as suspicious, Average Weight Drift (Avg. Drift) measures the average change in benign memory weights, and Attenuated denotes the fraction of benign memories whose weights are actually reduced. As shown in Table 6, DREAM achieves low FPRs of 2.98% and 0.66%, with Avg. Drift of only 0.0055 and 0.0009 on eICU and MIMIC-III, respectively. Moreover, only 1.62% and 1.05% of benign memories are attenuated. Together with the preserved ACC, these results show that DREAM introduces only limited changes to clean memory banks and avoids unnecessary forgetting of benign memories.

Table 6: Performance of DREAM on fully benign memory banks. The lower **FPR**, **Avg. Drift**, and **Atten.** are better (↓), while higher **ACC** is better (↑).

Dataset	ACC (↑)	FPR (↓)	Avg. Drift (↓)	Atten. (↓)
eICU	90.92	2.98	0.0055	1.62
MIMIC-III	74.89	0.66	0.0009	1.05

How does DREAM perform under different parameter settings? We vary σ, λ_0, k, and α_{rec} on EHRAgent (eICU) and ReAct with *GPT-5.4-mini*. Full results are reported in Appendix H Table 11. Fig. 5 demonstrates that EHRAgent is comparatively stable once perturbation probing is enabled, with the best ACC 85.03% achieved at $\sigma = 0.05$, $\lambda_0 = 0.4$, $k = 10$, and $\alpha_{rec} = 0.6$. In contrast, disabling key components degrades performance. For example, setting $\sigma = 0$, $k = 0$, and $\alpha_{rec} = 0$ reduces ACC to 63.64%, 65.78%, and 67.38%, respectively. ReAct is much more sensitive to parameter choice: its best ACC 61.20% is obtained at $\lambda_0 = 0.2$, $k = 15$, and $\alpha_{rec} = 0.5$, whereas $\sigma = 0$, $\lambda_0 = 0$, and $\alpha_{rec} = 0$ reduce ACC to 36.8%, 34.0%, and 38.0%, and overly large amplification ($k = 25$) further lowers ACC to 32.4%.



(a) EHRAgent (eICU) (b) ReAct

Figure 5: Performance comparison (ACC) of Ehragent and ReAct under different parameter settings.

DREAM remains effective in Adaptive Probe-Aware Attacks. We further evaluate a stronger adaptive attacker that jointly optimizes attack effectiveness and evasion of DREAM's diagnostic signals. As shown in Table 7, the adaptive setting substantially weakens existing defenses, with most baselines retaining high ASRs across retrieval, reasoning, and end-to-end stages. In contrast, DREAM reduces ASR-r and ASR-a to 12.46% and 14.32%, respectively, while achieving the highest ACC of 78.07%. These results show that DREAM maintains a clear robustness advantage even when the attacker explicitly adapts to its diagnostic mechanism.

Table 7: Defense performance against an adaptive probe-aware attacker. The lower **ASR-r**, **ASR-a**, and **ASR-t** are better (↓), while higher **ACC** is better (↑).

Method	ASR-r (↓)	ASR-a (↓)	ASR-t (↓)	ACC (↑)
No Defense	100.00	100.00	100.00	66.21
PPL	100.00	100.00	98.72	56.83
LLM Auditor	82.32	82.32	92.47	73.33
Distil Classifier	100.00	80.03	100.00	67.32
A-MemGuard	83.42	89.64	79.25	70.98
DREAM(Ours)	**12.46**	**14.32**	**12.46**	**78.07**

6 Conclusion

We presented DREAM, a neuro-inspired, agent-intrinsic memory repair framework for memory-poisoned LLM agents that performs selective functional forgetting without relying on expensive online auditing. By combining perturbation-based probing, multi-dimensional activation diagnosis, and selective forgetting of poison records, DREAM suppresses suspicious memories while preserving benign functionality. Experiments show that DREAM provides robust protection across most evaluated agent environments and remains efficient compared with baselines. Future work will focus on broader evaluations in more open-ended agent settings.

References

[1] T. Tu, M. Schaekermann, A. Palepu, K. Saab, J. Freyberg, R. Tanno, A. Wang, B. Li, M. Amin, Y. Cheng, *et al.*, "Towards conversational diagnostic artificial intelligence," *Nature*, vol. 642, no. 8067, pp. 442–450, 2025.

[2] W. Shi, R. Xu, Y. Zhuang, Y. Yu, J. Zhang, H. Wu, Y. Zhu, J. C. Ho, C. Yang, and M. D. Wang, "EHRAgent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Nov. 2024.

[3] J. Li, Y. Lai, W. Li, J. Ren, M. Zhang, X. Kang, S. Wang, P. Li, Y.-Q. Zhang, W. Ma, *et al.*, "Agent hospital: A simulacrum of hospital with evolvable medical agents," *arXiv preprint arXiv:2405.02957*, 2024.

[4] C. Cui, Z. Yang, Y. Zhou, Y. Ma, J. Lu, L. Li, Y. Chen, J. Panchal, and Z. Wang, "Personalized autonomous driving with large language models: Field experiments," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 20–27, IEEE, 2024.

[5] B. Wang, H. Cui, B. Zhang, X. Zheng, B. Guo, and Z. Yu, "Meco: Enhancing llm-empowered multi-robot collaboration via similar task memoization," *arXiv preprint arXiv:2601.20577*, 2026.

[6] J. Mao, J. Ye, Y. Qian, M. Pavone, and Y. Wang, "A language agent for autonomous driving," *arXiv preprint arXiv:2311.10813*, 2023.

[7] Z. Chen, Z. Xiang, C. Xiao, D. Song, and B. Li, "Agentpoison: Red-teaming llm agents via poisoning memory or knowledge bases," *Advances in Neural Information Processing Systems*, vol. 37, pp. 130185–130213, 2024.

[8] S. Dong, S. Xu, P. He, Y. Li, J. Tang, T. Liu, H. Liu, and Z. Xiang, "Memory injection attacks on LLM agents via query-only interaction," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[9] B. D. Sunil, I. Sinha, P. Maheshwari, S. Todmal, S. Mallik, and S. Mishra, "Memory poisoning attack and defense on memory based llm-agents," *arXiv preprint arXiv:2601.05504*, 2026.

[10] R. Shahroz, Z. Tan, S. Yun, C. Fleming, and T. Chen, "Agents under siege: Breaking pragmatic multi-agent llm systems with optimized prompt attacks," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9661–9674, 2025.

[11] S. Wang, T. Zhu, B. Liu, M. Ding, D. Ye, W. Zhou, and P. Yu, "Unique security and privacy threats of large language models: A comprehensive survey," *ACM Computing Surveys*, vol. 58, no. 4, pp. 1–36, 2025.

[12] X. Deng, Y. Zhang, J. Wu, J. Bai, S. Yi, Z. Zou, Y. Xiao, R. Qiu, J. Ma, J. Chen, *et al.*, "Taming openclaw: Security analysis and mitigation of autonomous llm agent threats," *arXiv preprint arXiv:2603.11619*, 2026.

[13] Z. Zhong, Z. Huang, A. Wettig, and D. Chen, "Poisoning retrieval corpora by injecting adversarial passages," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13764–13775, 2023.

[14] Y. Li, Q. Hu, Y. Zhang, L. Quan, J. Yu, and J. Wang, "Dynatrust: Defending multi-agent systems against sleeper agents via dynamic trust graphs," *arXiv preprint arXiv:2603.15661*, 2026.

[15] Q. Wei, T. Yang, Y. Wang, X. Li, L. Li, Z. Yin, Y. Zhan, T. Holz, Z. Lin, and X. Wang, "A-memguard: A proactive defense framework for llm-based agent memory," *arXiv preprint arXiv:2510.02373*, 2025.

[16] H. Luo, S. Dai, C. Ni, X. Li, G. Zhang, K. Wang, T. Liu, and H. Salam, "Agentauditor: Human-level safety and security evaluation for llm agents," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[17] G. Alon and M. Kamfonas, "Detecting language model attacks with perplexity," *arXiv preprint arXiv:2308.14132*, 2023.

[18] H. Inan, K. Upasani, J. Chi, R. Rungta, K. Iyer, Y. Mao, M. Tontchev, Q. Hu, B. Fuller, D. Testuggine, *et al.*, "Llama guard: Llm-based input-output safeguard for human-ai conversations," *arXiv preprint arXiv:2312.06674*, 2023.

[19] P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, and J. Z. Kolter, "Tofu: A task of fictitious unlearning for llms," 2024.

[20] M. Yu, F. Meng, X. Zhou, S. Wang, J. Mao, L. Pan, T. Chen, K. Wang, X. Li, Y. Zhang, *et al.*, "A survey on trustworthy llm agents: Threats and countermeasures," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 6216–6226, 2025.

[21] F. He, T. Zhu, D. Ye, B. Liu, W. Zhou, and P. S. Yu, "The emerged security and privacy of llm agent: A survey with case studies," *ACM Computing Surveys*, vol. 58, no. 6, pp. 1–36, 2025.

[22] J. A. Berry, D. C. Guhle, and R. L. Davis, "Active forgetting and neuropsychiatric diseases," *Molecular psychiatry*, vol. 29, no. 9, pp. 2810–2820, 2024.

[23] J. R. Wessel and M. C. Anderson, "Neural mechanisms of domain-general inhibitory control," *Trends in Cognitive Sciences*, vol. 28, no. 2, pp. 124–143, 2024.

[24] J. C. Hulbert, R. N. Henson, and M. C. Anderson, "Inducing amnesia through systemic suppression," *Nature Communications*, vol. 7, no. 1, p. 11003, 2016.

[25] Z. Zhu and Y. Wang, "Forgetting unrelated episodic memories through suppression-induced amnesia.," *Journal of Experimental Psychology: General*, vol. 150, no. 3, p. 401, 2021.

[26] Z. Zhu, M. C. Anderson, and Y. Wang, "Inducing forgetting of unwanted memories through subliminal reactivation," *Nature Communications*, vol. 13, no. 1, p. 6496, 2022.

[27] Q. Yang, Y. Wang, C. Liu, P. Zhi, S. Chen, Y. Lei, F. Xue, and Z. Zhu, "Reducing ptsd symptoms through unconscious intervention," *Proceedings of the National Academy of Sciences*, vol. 123, no. 11, p. e2521088123, 2026.

[28] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *The eleventh international conference on learning representations*, 2022.

[29] Z. Li, Y. Mi, Z. Zhou, H. Jiang, G. Zhang, K. Wang, and J. Fang, "Goal-aware identification and rectification of misinformation in multi-agent systems," *arXiv preprint arXiv:2506.00509*, 2025.

[30] Z. Xiang, F. Jiang, Z. Xiong, B. Ramasubramanian, R. Poovendran, and B. Li, "Badchain: Backdoor chain-of-thought prompting for large language models," in *The Twelfth International Conference on Learning Representations*, 2024.

[31] H. Lin, Y. Lao, T. Geng, T. Yu, and W. Zhao, "Uniguardian: A unified defense for detecting prompt injection, backdoor attacks and adversarial attacks in large language models," *arXiv preprint arXiv:2502.13141*, 2025.

[32] K. Wang, B. Zeng, Z. Wei, C. Jin, H. Zhou, X. Li, C. Yang, J. Qu, X. Xu, and X. Hu, "Trinityguard: A unified framework for safeguarding multi-agent systems," *arXiv preprint arXiv:2603.15408*, 2026.

[33] X. Li, R. Mao, Y. Zhang, R. Lou, C. Wu, and J. Wang, "Chain-of-scrutiny: Detecting backdoor attacks for large language models," in *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 7705–7727, 2025.

[34] 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.

[35] D. Ayzenshteyn, R. Weiss, and Y. Mirsky, "The best defense is a good offense: Countering llm-powered cyberattacks," *arXiv preprint arXiv:2410.15396*, 2024.

[36] H. Zhang, J. Huang, K. Mei, Y. Yao, Z. Wang, C. Zhan, H. Wang, and Y. Zhang, "Agent security bench (asb): Formalizing and benchmarking attacks and defenses in llm-based agents," in *13th International Conference on Learning Representations, ICLR 2025*, pp. 88011–88046, International Conference on Learning Representations, ICLR, 2025.

[37] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[38] F. Noé, S. Olsson, J. Köhler, and H. Wu, "Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning," *Science*, vol. 365, no. 6457, p. eaaw1147, 2019.

[39] W. Xu, Z. Liang, K. Mei, H. Gao, J. Tan, and Y. Zhang, "A-mem: Agentic memory for llm agents," *arXiv preprint arXiv:2502.12110*, 2025.

[40] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, and J. Berant, "Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 346–361, 2021.

[41] OpenAI, "Introducing gpt-5.4 mini and nano." `https://openai.com/index/introducing-gpt-5-4-mini-and-nano`, 2026. Accessed: 2026-04-24.

[42] Anthropic, "Claude haiku 4.5 system card," tech. rep., Anthropic, 2025.

[43] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," in *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

[44] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *International conference on machine learning*, pp. 3929–3938, PMLR, 2020.

[45] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

A Limitations

DREAM relies on a multi-dimensional activation probe to distinguish suspicious memories from benign ones. We have further evaluated a probe-aware adaptive attacker that explicitly optimizes poisoned memories against DREAM's diagnostic signals. Although DREAM remains substantially more effective than existing baselines under this stronger threat model, its attack success rates increase compared with the standard setting, indicating that adaptive optimization can partially weaken the diagnostic margin. This suggests that attackers with more complete knowledge of the defense mechanism, stronger optimization budgets, or joint adaptation to both diagnosis and memory-weight evolution may pose additional challenges. Further improving robustness against increasingly adaptive attackers remains an important direction for future work.

B Implementation Details

B.1 Hyperparameters

The hyperparameters for DREAM and our experiments are reported in Table 8.

Table 8: Hyperparameter Settings for DREAM

Parameters	Value
Temperature τ	0.05
The sharpness of coherence decay κ	2.00
Diagnostic sensitivity η	0.80
Suspect memory decay rate γ	0.50
Cold archive threshold θ_d	0.10
Utility Threshold θ_c	0.50
Iterations per forgetting step	10
Injection number	4

In addition to the above hyperparameters, for EHRAgent we set the noise standard deviation to $\sigma = 0.05$, the base decay rate to $\lambda_0 = 0.40$, the amplification factor to $k = 10$, and the benign recovery rate to $\alpha_{rec} = 0.60$. For ReAct, we set the noise standard deviation to $\sigma = 0.30$, the base decay rate to $\lambda_0 = 0.20$, the amplification factor to $k = 15$, and the benign recovery rate to $\alpha_{rec} = 0.50$. All experiments are conducted on a server equipped with six NVIDIA A100 GPUs (80 GB memory each).

B.2 Baselines

B.2.1 Distil Classifier

We adopt a lightweight binary text classifier based on `DistilBERT-base-uncased` [45] as a content-based defense baseline [34]. The model is fine-tuned to classify memory entries as *safe* or *harmful* according to their textual content alone. Training data is constructed from a composite safe/harmful prompt dataset and split into 80% training and 20% validation sets. Inputs are tokenized with `DistilBertTokenizer` and padded or truncated to 25 tokens. The model is trained for 10 epochs using AdamW (1×10^{-5}, batch size 32), with a `WeightedRandomSampler` to address class imbalance and gradient clipping at 1.0 for stability. The checkpoint with the lowest validation loss is used for inference.

B.2.2 Perplexity Filter Baseline

We adopt a perplexity-based filtering [17] baseline to detect anomalous memory entries according to their textual fluency and coherence. Perplexity scores are computed by a frozen `LLaMA-3.1-8B-Instruct` model, with all inputs padded or truncated to 100 tokens. For an

input sequence x, the score is defined as

$$\mathrm{PPL}(x) = \exp\left(-\frac{1}{T'}\sum_{t=1}^{T'}\log p(x_t \mid x_{<t})\right), \tag{14}$$

where T' is the number of non-padding tokens.

We use a two-stage filtering scheme. First, entries with $\mathrm{PPL}(x) > 10000$ are directly discarded as obvious anomalies. Second, for the remaining entries, we apply a dynamic threshold based on the median absolute deviation (MAD): a memory with perplexity p_i is filtered if

$$p_i > m + k \cdot \mathrm{MAD}, \tag{15}$$

where m is the median perplexity of the batch, MAD is the median absolute deviation, and $k = 1.0$ in our experiments.

B.2.3 LLM Auditor Baseline

We adopt an LLM-based auditing baseline that filters the retrieved memory context before it is used by the agent. In our implementation, `GPT-5.4-mini` and `Claude-Haiku-4.5` serve as the auditor models. The auditor is instructed to act as a security analyst that removes manipulative, toxic, or logically incoherent content while preserving legitimate information. Given a retrieved memory set $\mathcal{M}_{\mathrm{topK}}$, the auditor returns a cleaned memory context

$$\hat{\mathcal{M}} = \mathrm{Auditor}(\mathcal{M}_{\mathrm{topK}}), \tag{16}$$

which is then passed to the agent for downstream reasoning. This baseline represents an online memory-sanitization strategy.

B.2.4 A-MemGuard

A-MemGuard [15] is an external online defense framework for memory-augmented LLM agents. It protects the agent through two components: *consensus-based validation*, which detects anomalous memories by comparing parallel reasoning paths induced by different retrieved memories, and a *dual-memory structure*, which stores detected failures as lesson memories to guide future action revision. Given a retrieved memory set $\mathcal{M}_r$, A-MemGuard computes divergence scores

$$d_i = S_{\mathrm{div}}(\rho_i, \mathcal{P}_t), \tag{17}$$

where ρ_i is the reasoning path induced by memory m_i and $\mathcal{P}_t$ is the set of all parallel paths. Memories whose paths deviate from the consensus are filtered out, and anomalous paths are further distilled into lesson records for future correction. Thus, A-MemGuard represents a context-aware online auditing baseline with an additional external lesson memory for self-correction.

C Results under the REALM retriever.

Table 9 reports the defense performance when the retriever is replaced by REALM. Overall, the main trend remains consistent with the DPR-based results: DREAM is most effective on the structured EHRAgent tasks, while open-ended ReAct remains the more challenging case. Under *GPT-5.4-mini*, DREAM substantially improves robustness on both medical datasets. On eICU, it reduces ASR-r/a/t from $100.00/100.00/99.46$ to $2.14/2.14/1.60$, while also improving ACC from 74.80 to 78.07. On MIMIC-III, DREAM similarly lowers ASR-r/a/t to $2.42/1.82/0.00$ and raises ACC from 47.27 to 54.55. These results suggest that, even with a different retrieval backbone, DREAM continues to effectively suppress attack propagation in domain-specific agents.

On ReAct (Strategy-QA), the pattern is more nuanced. DREAM still achieves the lowest retrieval-stage and end-to-end attack rates under GPT-5.4-mini, reducing ASR-r from 78.33 to 0.48 and ASR-t from 88.78 to 2.00. However, ASR-a remains non-negligible (34.00), indicating that residual poisoned context may still propagate through ReAct's open-ended multi-step reasoning even after retrieval-stage exposure has been substantially reduced. This trend is also observed across several baselines in Table 9, suggesting that reasoning-stage robustness under REALM is more sensitive to the

downstream reasoning process and backbone. This remains a challenging regime for memory-level defenses and motivates future joint memory–reasoning protection.

Across methods, PPL and Distil Classifier remain weak on the EHRAgent settings, often leaving attack success near saturation. LLM Auditor and A-MemGuard provide partial mitigation in some cases, but their effectiveness is less consistent across datasets and backbones. Taken together, these results indicate that REALM alone does not eliminate memory poisoning risks; DREAM remains the most reliable defense on structured medical agents and provides the strongest overall suppression of retrieval and end-to-end attacks, while reasoning-stage defense for open-ended agents remains the harder regime.

Table 9: Performance evaluation of defense mechanisms against AgentPoison [7] attacks across different LLMs and datasets. Lower Attack Success Rate (ASR) in percentage (%) indicates better performance (↓), while higher benign accuracy (ACC) is preferable (↑).The retriver is **REALM**. For ASR, the minimum number in each column is in **bold**; for ACC, the maximum is in **bold**.

LLM	Defense	EHRAgent								ReAct			
		eICU				MIMIC-III				Strategy-QA			
		ASR-r	ASR-a	ASR-t	ACC	ASR-r	ASR-a	ASR-t	ACC	ASR-r	ASR-a	ASR-t	ACC
GPT-5.4-mini	No defense	100.00	100.00	99.46	74.80	100.00	92.73	23.03	47.27	78.33	93.81	88.78	34.39
	PPL	100.00	99.47	98.93	38.50	100.00	100.00	61.21	32.12	1.96	40.00	**1.00**	42.50
	LLM Auditor	73.26	67.91	94.12	50.80	99.39	100.00	15.15	39.39	0.95	80.00	2.00	47.00
	Distil Classifier	100.00	99.47	98.93	45.99	100.00	0.00	65.45	38.79	2.06	37.50	1.50	**50.50**
	A-MemGuard	64.17	72.19	53.48	44.92	72.12	**0.00**	15.38	18.27	4.44	50.00	3.08	46.55
	DREAM (ours)	**2.14**	**2.14**	**1.60**	**78.07**	**2.42**	1.82	**0.00**	**54.55**	**0.48**	**34.00**	2.00	36.50
Claude-haiku-4.5	No defense	100.00	86.10	59.89	42.25	100.00	72.12	13.33	39.39	94.30	72.07	60.85	23.58
	PPL	98.93	64.17	23.53	26.20	99.39	0.61	0.00	33.33	13.41	65.24	70.83	58.50
	LLM Auditor	100.00	89.84	47.06	32.62	61.82	0.00	7.27	36.36	9.01	60.30	6.53	62.36
	Distil Classifier	100.00	57.22	22.99	26.20	100.00	0.61	6.67	40.61	8.33	**60.00**	1.50	61.50
	A-MemGuard	68.98	37.43	5.35	28.34	52.63	**0.00**	0.00	36.84	8.42	67.00	1.50	62.50
	DREAM (ours)	**2.14**	**6.95**	**1.07**	**57.75**	**2.42**	3.63	**0.00**	**46.67**	**4.08**	75.00	**1.50**	**64.00**

D Noise distributions for subconscious probing.

Given a memory embedding $v_i \in \mathbb{R}^d$, subconscious probing perturbs it as

$$\tilde{v}_i = v_i + \epsilon_i, \tag{18}$$

where $\epsilon_i \in \mathbb{R}^d$ is sampled from a predefined noise distribution. In our ablation study, we consider the following four noise types.

Isotropic Gaussian noise. The perturbation is sampled independently and identically across all dimensions:

$$\epsilon_i \sim \mathcal{N}(0, \sigma^2 I), \tag{19}$$

where I is the identity matrix and $\sigma > 0$ controls the perturbation scale. This noise is *isotropic*, meaning that all directions in the embedding space are perturbed equally in expectation. As a result, it provides an unbiased probe of the local geometric stability around each memory embedding.

Uniform noise. The perturbation on each dimension is drawn from a bounded uniform distribution:

$$\epsilon_{i,j} \sim \mathbf{U}(-a, a), \qquad j = 1, \ldots, d, \tag{20}$$

where $a > 0$ is the noise range. To make it comparable with Gaussian perturbation, a can be chosen such that the variance matches, i.e.,

$$\mathrm{Var}[\epsilon_{i,j}] = \frac{a^2}{3} \approx \sigma^2. \tag{21}$$

Unlike Gaussian noise, uniform noise is bounded and does not emphasize rare larger perturbations.

Sparse Gaussian noise. Only a subset of dimensions is perturbed, while the others remain unchanged:

$$\epsilon_i = m_i \odot z_i, \tag{22}$$

where $\odot$ denotes element-wise multiplication, $z_i \sim \mathcal{N}(0, \sigma^2 I)$, and the mask $m_i \in \{0, 1\}^d$ is sampled as

$$m_{i,j} \sim \mathrm{Bernoulli}(p). \tag{23}$$

Here $p \in (0, 1]$ is the activation probability of each dimension. Sparse Gaussian noise probes only localized coordinates of the embedding, producing a more selective but less globally informative perturbation.

Anisotropic Gaussian noise. The perturbation follows a Gaussian distribution with direction-dependent covariance:

$$\epsilon_i \sim \mathcal{N}(0, \Sigma), \tag{24}$$

where $\Sigma \in \mathbb{R}^{d \times d}$ is a positive semi-definite covariance matrix. A common special case is diagonal anisotropy,

$$\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2), \tag{25}$$

which assigns different perturbation magnitudes to different dimensions. Compared with isotropic Gaussian noise, anisotropic Gaussian noise emphasizes certain directions in the embedding space and may therefore introduce directional bias in stability probing. For a fair comparison, different noise types are calibrated to have approximately matched expected perturbation energy, i.e.,

$$\mathbb{E}\|\epsilon_i\|_2^2 \approx d\sigma^2. \tag{26}$$

Interpretation. These four perturbations differ in how they explore the local neighborhood around a memory embedding. Isotropic Gaussian noise perturbs all directions evenly and thus provides the most geometry-preserving probe of local stability. Uniform noise is similarly dense but bounded. Sparse Gaussian noise perturbs only partial coordinates and may miss global fragility patterns. Anisotropic Gaussian noise is dense but directionally biased, which can overemphasize certain subspaces. This difference helps explain why isotropic Gaussian noise yields the strongest benign utility in our ablation while maintaining consistently low attack success.

Table 10: Performance of different noise type.

Noise Type	**ASR-r**	**ASR-a**	**ASR-t**	**ACC**
Isotropic Gussian	2.14	2.14	2.14	**85.03**
Uniform	2.14	2.14	2.14	77.01
Sparse Gaussian	2.14	2.67	2.14	77.54
Anisotropic Gussian	2.14	2.14	2.14	81.28

As shown in Table 10, all tested noise types keep the attack success rates at a comparably low level, indicating that the overall defense mechanism is not overly sensitive to the specific perturbation family. However, their impact on benign performance differs more clearly. Isotropic Gaussian noise achieves the best ACC (85.03%), outperforming Uniform noise (77.01%), Sparse Gaussian noise (77.54%), and Anisotropic Gaussian noise (81.28%). In particular, Sparse Gaussian noise also slightly increases ASR-a to 2.67%, suggesting that overly localized perturbations provide a less reliable probing signal. These results support our choice of isotropic Gaussian noise, which offers the best balance between preserving benign utility and maintaining consistently low attack success.

E Analysis for $R_i(t)$

As shown in Fig. 6, the three components exhibit clearly different discriminative behaviors. The $\mathrm{sim}(\tilde{v}_i, v_i)$ distribution of benign and poisoned memories remains heavily overlapped, indicating that local geometric stability alone is insufficient for reliable diagnosis. In contrast, p_{ret} and E_{coh} show much stronger distributional shifts, suggesting that poisoned memories are less competitive in retrieval and weaker in semantic coherence under perturbation. Importantly, however, these two components capture different aspects of abnormality: p_{ret} reflects global retrieval competitiveness, whereas E_{coh} reflects semantic consistency. As further shown in Fig. 6, after aggregating these heterogeneous cues into the multi-dimensional activation intensity $R_i(t)$, the separation between benign and poisoned memories becomes substantially clearer than in any individual weak component such as $\mathrm{sim}(\tilde{v}_i, v_i)$. Therefore, as also demonstrated in Table 5, the key observation is not that any single component is universally sufficient, but that the three components provide heterogeneous yet complementary evidence. This precisely justifies the definition of $R_i(t)$ as a unified and more robust diagnostic score.

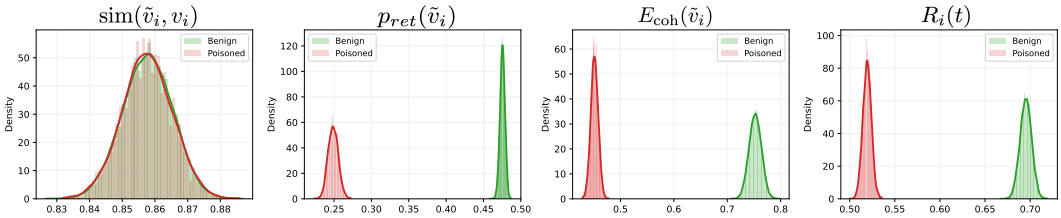


Figure 6: Empirical distributions of the three component signals used in $R_i(t)$. While $\mathrm{sim}(\tilde{v}_i, v_i)$ shows substantial overlap between benign and poisoned memories, p_{ret} and E_{coh} exhibit clearer shifts. This heterogeneous behavior motivates their aggregation into the multi-dimensional activation intensity $R_i(t)$.

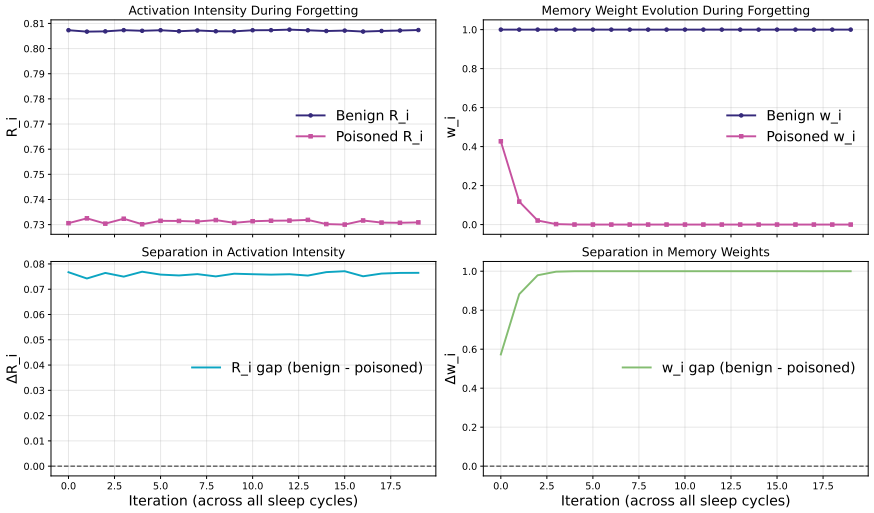


Figure 7: Memory forgetting dynamics of benign and poisoned memories across repeated sleep cycles. The top row shows the evolution of activation intensity $R_i(t)$ and memory weight $w_i(t)$, respectively. The bottom row reports the corresponding separation gaps between benign and poisoned memories. While the $R_i(t)$ gap remains consistently positive, the $w_i(t)$ gap rapidly expands and saturates, indicating that stable activation-level differences are progressively translated into stronger functional separation during forgetting.

F Forgetting dynamics of R_i and w_i.

Fig. 7 visualizes the evolution of activation intensity and memory weights during repeated sleep cycles. As shown in the top-left panel, benign memories consistently maintain higher $R_i(t)$ values than poisoned ones, and this gap remains stably positive throughout the entire forgetting process. This indicates that the proposed multi-dimensional activation intensity provides a persistent diagnostic signal rather than a transient fluctuation. In turn, the top-right panel shows that this stable discrepancy in $R_i(t)$ is translated into sharply different weight trajectories: benign memories remain close to unit weight, whereas poisoned memories are rapidly attenuated toward zero within only a few iterations. The bottom panels further highlight this effect quantitatively. While the $R_i(t)$ gap stays relatively constant across cycles, the $w_i(t)$ gap quickly increases and saturates near its maximum, showing that DREAM converts a stable activation-level separation into an increasingly pronounced functional separation in memory influence. Together, these results illustrate the intended behavior of selective functional forgetting: suspicious memories are progressively suppressed, whereas benign memories are stably preserved.

G Online Consolidation Signal and Memory-level Utility Attribution

Online Consolidation Signal. Let $c_i(t) \in [0, 1]$ denote the online consolidation signal of memory v_i, which reflects its historical cross-task utility during the agent's online phase. Intuitively, $c_i(t)$ serves as a long-term trust indicator: memories that are repeatedly retrieved and associated with successful task completion should be less aggressively attenuated during subsequent repair cycles.

To avoid overreacting to transient task-specific fluctuations, we update the consolidation signal using an exponential moving average. Specifically, for online cycle t and query q_t, let $\mathcal{M}_{\mathrm{topK}}(q_t)$ denote the retrieved memory set. We update $c_i(t)$ only when v_i is actually retrieved:

$$c_i(t) = \begin{cases} \delta\, c_i(t-1) + (1-\delta)\, u_i(t), & v_i \in \mathcal{M}_{\mathrm{topK}}(q_t), \\ c_i(t-1), & \text{otherwise}, \end{cases} \tag{27}$$

where $\delta \in (0,1)$ is a momentum coefficient controlling temporal smoothness.

Here, $u_i(t) \in [0,1]$ denotes the memory-level utility assigned to v_i at online cycle t. Since multiple retrieved memories may jointly contribute to the final task outcome, we adopt a lightweight attribution scheme that decomposes task-level utility into memory-level contributions. Let $U_t \in [0,1]$ denote the task-level utility signal achieved at cycle t (i.e., task success signal). We define

$$u_i(t) = a_i(t) \cdot U_t, \tag{28}$$

where $a_i(t)$ is a normalized attribution weight over the retrieved set:

$$a_i(t) = \begin{cases} \dfrac{\exp(\mathrm{Score}(q_t, v_i)/\tau_u)}{\sum_{v_j \in \mathcal{M}_{\mathrm{topK}}(q_t)} \exp(\mathrm{Score}(q_t, v_j)/\tau_u)}, & v_i \in \mathcal{M}_{\mathrm{topK}}(q_t), \\ 0, & \text{otherwise}. \end{cases} \tag{29}$$

Here, $\mathrm{Score}(q_t, v_i)$ denotes the retrieval score of memory v_i under query q_t, and $\tau_u > 0$ is a temperature parameter. This formulation assigns larger utility to memories that are both retrieved and more strongly implicated in successful task completion, while avoiding the computational overhead of leave-one-out re-evaluation.

Overall, the online consolidation signal $c_i(t)$ provides a complementary utility-aware factor to the offline anomaly diagnosis: suspicious memories with persistently low utility remain vulnerable to attenuation, whereas benign but atypical memories can recover long-term trust through repeated successful use.

H Detailed Sensitivity Analysis on Defense Parameters

To provide a more complete view of DREAM's internal behavior, Table 11 reports the full parameter sensitivity results for σ, λ_0, k, and α_{rec} on both EHRAgent and ReAct.

Effect of perturbation scale (σ). The perturbation scale controls the strength of subconscious probing. On EHRAgent, setting $\sigma = 0$ causes a near-complete defense collapse (ASR-r/a/t $= 99.47/99.47/99.47$), showing that perturbation probing is indispensable. Once probing is enabled, the attack rates remain uniformly low (all 2.14%), while ACC varies within a narrower range and peaks at 85.03% when $\sigma = 0.05$. ReAct is substantially more sensitive to σ: small perturbations often fail to suppress reasoning-stage attacks, whereas intermediate values such as $\sigma = 0.3$ and $\sigma = 0.5$ reduce all ASRs to zero and achieve the best ACC values of 61.20% and 53.20%, respectively. This suggests that probing is necessary for both agents, but its effective scale is much more delicate for open-ended reasoning.

Effect of base decay rate (λ_0). The parameter λ_0 sets the baseline attenuation strength. On EHRAgent, moderate values (0.1–0.6) keep ASRs low, with the best ACC of 85.03% achieved at $\lambda_0 = 0.4$. However, overly weak or poorly calibrated decay can still be harmful: at $\lambda_0 = 0$, defense collapses completely, while at $\lambda_0 = 0.7$ and 0.8 the attack rates rise again to near-saturation. On ReAct, λ_0 is the most sensitive parameter. Several values (e.g., 0.2, 0.4, 0.8, and 0.9) suppress all ASRs to zero, but the corresponding ACC varies widely, from 36.61% to 61.20%. The best configuration is $\lambda_0 = 0.2$, indicating that decay calibration is critical for preserving utility in open-ended agents.

Effect of amplification factor (k). The factor k determines how strongly suspect scores amplify attenuation. EHRAgent is relatively stable in ASR across all tested k, but ACC is clearly optimized by moderate amplification, peaking at 85.03% when $k = 10$. Setting $k = 0$ lowers ACC to 65.78%, showing that amplification contributes meaningfully to utility-preserving filtering. ReAct again

Table 11: Parameter sensitivity analysis of DREAM on EHRAgent (eICU) and ReAct. All metrics are reported in percentages (%). The lower ASR is better (↓), while the higher ACC is better (↑).

Parameter	Value	EHRAgent (eICU)				ReAct			
		ASR-r	ASR-a	ASR-t	ACC	ASR-r	ASR-a	ASR-t	ACC
σ	0.00	99.47	99.47	99.47	63.64	0.42	100.00	1.60	36.80
	0.05	2.14	2.14	2.14	**85.03**	0.45	100.00	1.60	42.80
	0.10	2.14	2.14	2.14	75.94	0.32	100.00	1.20	50.00
	0.15	2.14	2.14	2.14	77.01	0.91	80.00	1.60	34.40
	0.20	2.14	2.14	2.14	81.82	0.40	100.00	1.60	39.60
	0.25	2.14	2.14	2.14	79.14	0.42	100.00	1.60	46.40
	0.30	2.14	2.14	2.14	77.54	0.00	0.00	0.00	**61.20**
	0.35	2.14	2.14	2.14	77.33	0.00	0.00	20.40	42.80
	0.40	2.14	2.14	2.14	77.01	0.32	100.00	1.20	41.29
	0.45	2.14	2.14	2.14	75.40	0.36	100.00	1.60	40.80
	0.50	2.14	2.14	2.14	77.54	0.00	0.00	0.00	53.20
λ_0	0.00	100.00	100.00	100.00	74.87	0.00	0.00	0.00	34.00
	0.10	2.14	2.14	2.14	79.14	0.40	100.00	1.60	44.80
	0.20	2.14	2.14	2.14	68.98	0.00	0.00	0.00	**61.20**
	0.30	3.74	3.74	3.74	74.33	0.41	100.00	1.60	42.80
	0.40	2.14	2.14	2.14	**85.03**	0.00	0.00	0.00	38.80
	0.50	2.14	2.14	2.14	76.47	0.43	100.00	1.60	44.80
	0.60	2.14	2.14	2.14	79.14	0.36	100.00	1.60	31.20
	0.70	100.00	98.93	98.93	72.19	0.39	100.00	0.40	34.00
	0.80	100.00	98.93	98.93	77.54	0.00	0.00	0.00	36.61
	0.90	2.14	0.534	0.534	69.52	0.00	0.00	0.00	54.40
	1.00	2.14	2.14	2.14	70.59	0.42	100.00	1.60	44.00
k	0	2.14	2.14	2.14	65.78	0.00	0.00	0.00	52.00
	5	2.14	2.14	2.14	80.74	0.43	100.00	1.60	44.80
	10	2.14	2.14	2.14	**85.03**	0.00	0.00	0.00	46.99
	15	2.14	2.14	2.14	78.07	0.00	0.00	0.00	**61.20**
	20	2.14	2.14	2.14	74.33	0.42	100.00	1.60	36.40
	25	2.14	2.14	2.14	72.74	1.48	28.57	0.80	32.40
α_{rec}	0.00	2.14	2.14	2.14	67.38	0.37	100.00	1.60	38.00
	0.10	2.14	2.14	2.14	77.01	0.48	100.00	2.00	46.00
	0.20	2.14	2.14	2.14	74.87	0.00	0.00	0.00	55.20
	0.30	2.14	2.14	2.14	73.80	0.00	0.00	0.00	41.20
	0.40	2.14	2.14	2.14	73.26	0.00	0.00	0.00	36.80
	0.50	2.14	2.14	2.14	73.28	0.00	0.00	0.00	**61.20**
	0.60	2.14	2.14	2.14	**85.03**	0.52	100.00	1.60	56.80
	0.70	2.14	2.14	2.14	73.26	0.71	40.00	0.80	38.80
	0.80	2.14	2.14	2.14	71.12	0.00	0.00	0.00	46.40
	0.90	2.14	2.14	2.14	66.84	0.42	100.00	1.60	44.40

exhibits a more delicate trade-off: moderate settings such as $k = 10$ and 15 drive all ASRs to zero, with the best ACC of 61.20% obtained at $k = 15$, whereas an overly large amplification ($k = 25$) sharply degrades ACC to 32.40% and also reintroduces nonzero attack success. This confirms that aggressive amplification can over-prune useful reasoning context in open-ended agents.

Effect of recovery rate (α_{rec}). The recovery rate controls the reconsolidation of benign but temporarily attenuated memories. On EHRAgent, ASRs remain unchanged across α_{rec}, but ACC is strongly affected: disabling recovery ($\alpha_{rec} = 0$) reduces ACC to 67.38%, while the best performance is achieved at $\alpha_{rec} = 0.6$ with 85.03%. On ReAct, recovery is even more consequential. Several intermediate settings ($\alpha_{rec} = 0.2$, 0.3, 0.4, 0.5, and 0.8) suppress all ASRs to zero, but ACC varies substantially, peaking at 61.20% when $\alpha_{rec} = 0.5$. When recovery is removed ($\alpha_{rec} = 0$), ACC drops to 38.00% and ASR-a returns to 100%. These results indicate that reconsolidation is not merely auxiliary, but essential for restoring benign functionality after attenuation, especially in agents with more fragile reasoning trajectories.

I Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 62372378, and the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University. We also sincerely thank the anonymous reviewers for their constructive comments and valuable suggestions, which helped improve the quality of this work.

J Use of Large Language Models (LLMs)

Large language models were used exclusively for linguistic polishing, such as improving grammar and readability. They were not involved in the formulation of research ideas, study design, implementation, data analysis, or reference selection. All scientific contributions, technical claims, and interpretations in this paper were conceived, authored, and rigorously validated by the human authors.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the abstract and introduction, we have thoroughly detailed the background, motivation, scope, main experimental results, and contributions of our work.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide discussions the limitation of our method in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The assumptions in this paper are properly referenced.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper has disclosed all the information in the method, experiment section and appendix. The data and code are also publicly accessible and provided with detailed procedures to reproduce the results reported in this paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and data are publicly accessible.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Details of both training and test are disclosed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines: We adopt standard evaluation datasets, LLM agents, and metrics, which are accompanied by statistical significance.

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The experiments are sufficiently discussed to be run by others.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The conducted research conforms in every respect with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Both potential positive societal impacts and negative societal impacts of the work have been discussed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: The paper has discussed safeguards for responsible release of data or models that have a high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the mentioned previous work are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The code will be made public upon acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: This research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.