Hallucination-Guided Unlearning: Using Hallucination Traces to Reveal Overfitted Memories

Jiayu Zhang[1*] **Ziqi Zhong**[2*] **Yuliang Gai**[3,4] **Canran Xiao**[5†]
[1]Peking University [2]UTS Business School, University of Technology Sydney
[3]Massachusetts Institute of Technology [4]University of Massachusetts Boston
[5]Shenzhen Campus of Sun Yat-sen University

Abstract

Selective unlearning is increasingly important for the safety and compliance of large language models. Harmful or memorized content is often entangled with useful knowledge and difficult to enumerate. We use the model's own failures to identify candidates for unlearning. We propose HALLUCINATION-GUIDED UNLEARNING (HGU), which uses hallucinations as diagnostic signals. HGU localizes unsupported spans via evidence and internal consensus, traces a sparse set of high-responsibility MLP neurons, and edits them with compact gates under a utility-preserving KL constraint. On TOFU, HGU achieves the best MU–FE trade-off, with Avg scores of 0.7221 (forget05) and 0.7699 (forget10), and the highest SepS H-Avg (0.4382/0.4715) under mixed forget/retain prompts. On the WMDP auxiliary task, HGU yields the strongest forgetting (e.g., 14.2 on Economics) while improving Retain (52.7) and MMLU (57.2). These results suggest that tracing hallucinations enables targeted, post-hoc unlearning that better balances forgetting efficacy and general utility without full retraining.

1 Introduction

Large language models (LLMs) are increasingly deployed as knowledge-intensive interfaces for search, coding, education, and decision support, where users expect both high utility and responsible behavior [Xiao et al., 2026b,c]. Medical decision-making and financial trading, for example, have motivated safeguards that check the evidence used by LLM agents [Hu, 2026, Wang and Saxena, 2026]. Yet training pipelines inevitably imprint undesirable content into model parameters—ranging from private or copyrighted text to hazardous know-how—making selective removal a first-class requirement for safety, compliance, and trustworthy deployment [Bourtoule et al., 2021, Li et al., 2024]. Because memorized influence can be diffuse and entangled with legitimate knowledge, unlearning must balance effective forgetting against broad capability preservation under tight compute constraints [Maini et al., 2024, Shi et al., 2025, Yang et al., 2026a].

Recent work has refined unlearning objectives and interventions and developed benchmarks for measuring forgetting–utility trade-offs [Maini et al., 2024, Li et al., 2024, Shi et al., 2025]. Most approaches optimize removal given an *explicit target*, such as a curated forget set, task vector, or labeled harmful corpus [Zhang et al., 2024, Ji et al., 2024, Gao et al., 2024, Hu et al., 2025, Shen et al., 2025, Zhang et al., 2025, Zhao et al., 2025]. When memorized content is *latent*, triggered only by certain prompts, or mixed with benign facts, unlearning also requires *discovering what to forget* and isolating the responsible internal pathways [Shi et al., 2025, Li et al., 2024]. Confident hallucinations can reveal overfitted memories that lack evidential support. We use these generation failures to guide unlearning.

*Equal contribution.
†Corresponding author: xiaocr3@mail.sysu.edu.cn.

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

? This paper asks:

Can hallucinations help us find what the model has memorized incorrectly, and remove it without retraining?

We compare generated content with trusted evidence and measure consensus across samples to localize hallucinated spans. We trace those spans to a sparse set of memory pathways and edit the pathways under a utility-preserving constraint.

- **(i)** We formulate unsupported-span-guided unlearning, using hallucinated spans as diagnostic probes.
- **(ii)** We propose HGU, a sparse post-hoc editing method that combines evidence-based span localization, exact local activation-gradient tracing, and utility-preserving gated intervention.
- **(iii)** We report leakage-free protocols and attribution implementation details. We evaluate on TOFU (including the SepS mixed-prompt protocol), MUSE, WMDP, and PKU-SafeRLHF, and report seed variability, compute costs, and quantization robustness.

Scope. HGU uses unsupported spans to identify candidates for reference- or policy-conditioned unlearning. Selecting accurate content for removal requires a policy or provenance signal. Missing knowledge and retrieval errors may call for correction. The evidence needed depends on the task, as when pointing gestures help disambiguate language in embodied reference resolution [Li et al., 2023]. The selected channels serve as intervention points, with the same knowledge potentially represented elsewhere. Appendix F reports blind discovery, independent evaluation, and re-elicitation tests for approximate post-hoc unlearning.

2 Related Work

LLM unlearning and evaluation. In federated learning, training-time influence logging facilitates client-level deletion [Xiao et al., 2026a]. LLM unlearning is commonly posed as removing the influence of a specified *forget set* while preserving utility on *retain* data and downstream tasks, extending classical machine unlearning to generative models [Bourtoule et al., 2021]. Recent benchmarks clarify this objective and reveal gaps in practice: TOFU tests fictitious profile forgetting [Maini et al., 2024], MUSE evaluates real-corpus removal across memorization, privacy, and deployment constraints [Shi et al., 2025], and WMDP frames unlearning as safety-motivated suppression of hazardous knowledge [Li et al., 2024]. Algorithmically, prior work refines "forget-vs.-retain" objectives to avoid collapse (e.g., NPO and related gradient-based or multi-objective variants) [Zhang et al., 2024, Maini et al., 2024, Pan et al., 2025], while logit-space approaches such as ULD edit behavior via learned logit differences [Ji et al., 2024]. Most methods assume explicit targets. Their performance degrades when harmful influence is latent, entangled, or triggered only by particular prompts, a limitation highlighted by broad, sequential evaluations [Shi et al., 2025]. We address this upstream bottleneck by using hallucinations to *identify* what to forget and performing sparse, trace-guided edits under an explicit utility-drift constraint.

Mechanistic and targeted editing. Recent work pursues *fine-grained* LLM unlearning by editing specific representations or parameter subspaces. Representative approaches include RMU for safety-focused representation manipulation [Li et al., 2024], activation/subspace methods such as LUNAR, FALCON, UNLEARN, and Ethos [Shen et al., 2025, Hu et al., 2025, Lizzo and Heck, 2025, Gao et al., 2024], and parameter-selection strategies like selective pruning (LLM-Eraser) and influence reweighting (RapidUn) [Zhang et al., 2025, Zhao et al., 2025]. While these methods clarify *where* and *how* to intervene, they typically rely on curated forget data or task vectors to specify *what* to remove. We use hallucination traces to localize hidden memorized content and apply sparse, utility-constrained gating to the responsible MLP pathways.

Evaluating selectivity. MUSE measures verbatim reproduction and knowledge recall separately [Shi et al., 2025], and WMDP probes hazardous knowledge through multiple-choice questions [Li et al., 2024]. Open-ended prompts offer a complementary test because a single response can contain both material to forget and material to retain. The SepS protocol used here combines forget and retain requests in a single prompt, testing whether an edit suppresses the target content while preserving other information in the same response.

3 Method

3.1 Preliminary Study

We run a preliminary diagnostic on a frozen aligned LLM. For each open-ended prompt x, we retrieve trusted evidence $e(x)$, sample multiple responses, and decompose each response into token-level or short phrase-level units. For every unit, we compute evidence support, cross-sample consensus, logit-margin confidence, and MLP attribution concentration. We then plot the units in an evidence–consensus phase map: color shows attribution concentration, contours show the unsupported-token rate, and point size shows model confidence. The phase map shows whether unsupported, low-consensus units also exhibit concentrated internal attribution. It distinguishes uncertain from confident unsupported tokens. Confident unsupported tokens are more relevant to overfitted or memorized behavior. The diagnostic measures support, stability, and attribution concentration before span or neuron selection, gating, and unlearning. Detailed metric definitions are given in Appendix D.1.

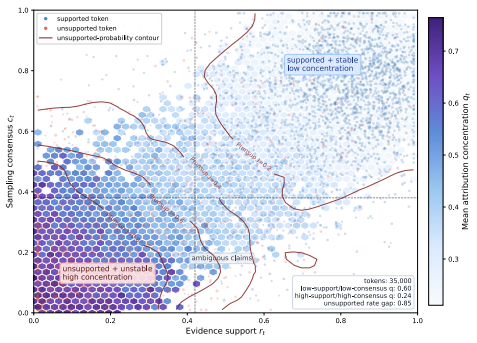

Figure 1: Unsupported and unstable generations concentrate in a high-concentration, high-confidence region.

Findings. Figure 1 reveals three patterns. First, unsupported tokens cluster where evidence support and sampling consensus are both low, indicating that external grounding and internal stability are complementary diagnostics (cf. structural and behavioral evidence fusion in cyber threat attribution Wang et al., 2026). Second, many unsupported tokens in this region have large logit margins. Third, attribution concentration rises sharply in the same region, suggesting that hallucinated claims are often driven by sparse internal pathways. These patterns suggest that hallucination traces contain a structured subset of externally unsupported and internally localized tokens. This motivates using low-support, low-consensus, high-concentration tokens as diagnostic probes for the trace-guided unlearning procedure below.

3.2 Overview

Hallucination-Guided Unlearning (HGU) uses *diagnostic traces* from hallucinations to identify high-responsibility pathways and suppresses them with sparse trainable neuron gates on a frozen backbone. Starting from a base model f_θ, HGU (i) selects hallucination-prone prompts and samples generations, (ii) localizes hallucinated spans via evidence support and stochastic-consensus, (iii) extracts neuron-level responsibility scores from MLP activations and their gradients, then (iv) trains sparse gates on the identified neurons to suppress hallucination-specific pathways while distilling the original behavior on a utility set. Figure 2 shows the pipeline.

Notation and Model Internals. Let f_θ be an autoregressive LLM with vocabulary $\mathcal{V}$ and conditional distribution $p_\theta(y \mid x) = \prod_{t=1}^{T} p_\theta(y_t \mid x, y_{<t})$. We analyze a standard Transformer with L layers. For layer l and position t, let $A_{l,h,t,\cdot}$ be the attention distribution of head h at query position t. Let $a_{l,t} \in \mathbb{R}^{d_{\text{ff}}}$ denote the post-nonlinearity hidden activation of the MLP block at layer l and position t (the FFN hidden width is d_{ff}). We compute all internal quantities using a single teacher-forced forward pass on the full sequence (x, y).

3.3 Hallucination Probing

Hidden memorized content rarely exposes itself under typical prompts. HGU therefore constructs a probe set that increases the rate of hallucination, treating hallucinated generations as stress tests that surface overconfident, non-evidence-grounded continuations.

Given a query pool $\mathcal{Q}$, we retrieve supporting evidence for each prompt x from a trusted corpus $\mathcal{K}$: $e(x) = \text{Retrieve}(x; \mathcal{K})$. We then score prompts using a hallucination propensity that increases with model uncertainty and decreases with evidence support. Let $\mathcal{H}_\theta(x) = -\sum_{w \in \mathcal{V}} p_\theta(w \mid x) \log p_\theta(w \mid$

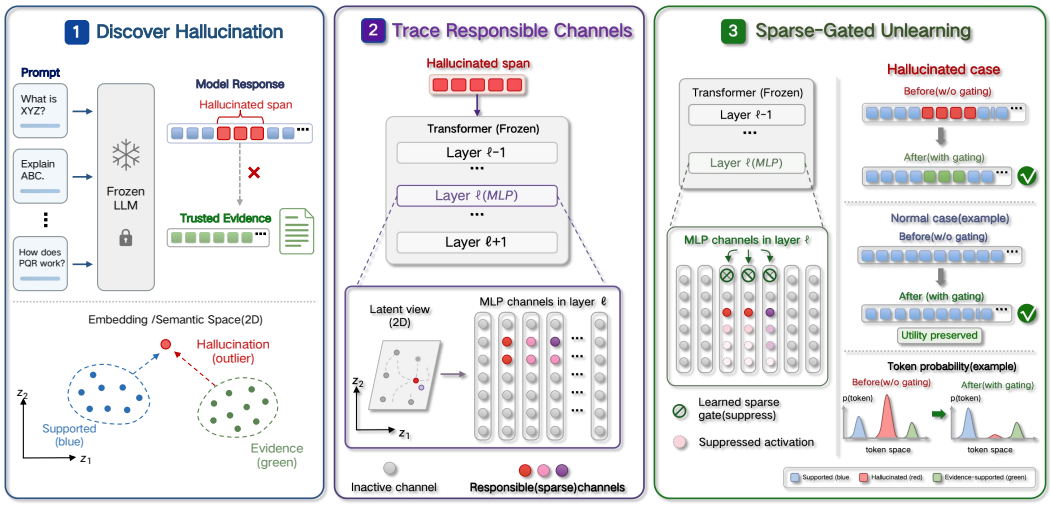

Figure 2: **Overview of Hallucination-Guided Unlearning.** HGU first discovers unsupported hallucinated spans by contrasting model responses with trusted evidence, then traces these spans to sparse high-responsibility MLP channels, and finally learns sparse gates to suppress hallucination-specific pathways while preserving normal evidence-supported behavior.

x) be the next-token entropy at the first generation step, and let $\mathrm{Cov}(x, e(x)) \in [0, 1]$ be a deterministic evidence coverage score defined in Appendix A.1. We define

$$\pi(x) \triangleq \alpha_\pi \, \mathcal{H}_\theta(x) \; + \; \beta_\pi \left(1 - \mathrm{Cov}(x, e(x))\right), \tag{1}$$

where $\alpha_\pi \geq 0$ and $\beta_\pi \geq 0$ are scalars. We select the top-B prompts $\mathcal{X} = \mathrm{TopB}_{x \in \mathcal{Q}} \pi(x)$ and sample one completion per prompt via stochastic decoding, yielding probe pairs $\mathcal{D}_{\mathrm{H}} = \{(x, y)\}_{x \in \mathcal{X}}$.

3.4 Span-Level Hallucination Localization

Unlearning requires localized targets. HGU assigns each generated token a hallucination weight that rises when the token lacks evidential support and remains unstable under stochastic-consensus at the same prefix.

Fix a probe pair $(x, y) \in \mathcal{D}_{\mathrm{H}}$. For each position t, we compute two scalar signals:

Evidence support. Let s_t be a short text span around y_t (Appendix A.3). A verifier v_ψ maps $(s_t, e(x))$ to an entailment-like score in $[0, 1]$. We set $\phi_{\mathrm{ref}}(x, y, t) = v_\psi(s_t, e(x))$, where larger values indicate stronger support.

Stochastic-consensus at fixed prefix. We run K stochastic forward passes of f_θ at the same prefix $(x, y_{<t})$, yielding next-token distributions $p_\theta^{(k)}(\cdot \mid x, y_{<t})$ for $k \in [K]$ (§A.4). We define the consensus score as the mean probability assigned to the realized token y_t: $\bar{p}_t \triangleq \frac{1}{K} \sum_{k=1}^{K} p_\theta^{(k)}(y_t \mid x, y_{<t})$, and use disagreement $\phi_{\mathrm{con}}(x, y, t) \triangleq 1 - \bar{p}_t$.

Hallucination weight and span extraction. We combine the two signals into a single hallucination weight

$$h_t \triangleq \sigma\Big(a \cdot \big(1 - \phi_{\mathrm{ref}}(x, y, t)\big) + b \cdot \phi_{\mathrm{con}}(x, y, t) - c\Big), \tag{2}$$

where $\sigma(u) = \frac{1}{1+e^{-u}}$ is the sigmoid, and $a \geq 0$, $b \geq 0$, $c \in \mathbb{R}$ are scalars. We mark hallucinated indices by $\mathcal{T}_{\mathrm{H}}(x, y) = \{t : h_t \geq \tau\}$ with threshold $\tau \in (0, 1)$ and extract maximal contiguous spans $\mathcal{S}(x, y)$ from $\mathcal{T}_{\mathrm{H}}(x, y)$.

3.5 From Hallucination Traces to Responsible Neurons

A hallucinated span is an output symptom. HGU turns this symptom into an internal target by identifying a sparse set of MLP neurons that (i) are highly active during hallucinated tokens and (ii)

have large local influence on the log-probabilities of those tokens, with an additional weight that emphasizes low-entropy attention states associated with overconfident generation.

We evaluate internals on (x, y) using teacher forcing. For each layer l and hallucinated position $t \in \mathcal{T}_{\mathrm{H}}(x, y)$, define the token log-probability $\ell_t(\theta) \triangleq \log p_\theta(y_t \mid x, y_{<t})$ and the activation gradient $g_{l,t} \triangleq \nabla_{a_{l,t}} \ell_t(\theta) \in \mathbb{R}^{d_{\mathrm{ff}}}$. We compute an attention-entropy weight $w_{l,t} \geq 0$ from $A_{l,h,t,\cdot}$ (§A.5), where larger $w_{l,t}$ indicates lower attention entropy.

We score each MLP neuron $i \in [d_{\mathrm{ff}}]$ at layer l by aggregating hallucination-weighted, entropy-weighted gradient–activation attribution:

$$I_{l,i} \triangleq \sum_{t \in \mathcal{T}_{\mathrm{H}}(x,y)} h_t \cdot w_{l,t} \cdot |a_{l,t,i}\, g_{l,t,i}|\,, \tag{3}$$

where $a_{l,t,i}$ and $g_{l,t,i}$ denote the i-th coordinate of $a_{l,t}$ and $g_{l,t}$, respectively. We select a fixed fraction of neurons per layer:

$$\mathcal{C}_l \triangleq \mathrm{Top}_{\lceil \rho d_{\mathrm{ff}} \rceil}\Big(\{I_{l,i}\}_{i=1}^{d_{\mathrm{ff}}}\Big)\,, \qquad \mathcal{C} \triangleq \bigcup_{l=1}^{L} \{(l, i) : i \in \mathcal{C}_l\}, \tag{4}$$

where $\rho \in (0, 1)$ is the sparsity ratio. The set $\mathcal{C}$ is the hallucination memory channel set used by the unlearning intervention.

3.6 Sparse-Gated Unlearning

HGU trains a small set of neuron gates on $\mathcal{C}$ with the backbone frozen. The gates suppress hallucination-responsible neurons while distilling the original model behavior on a utility set, which stabilizes general performance.

We introduce trainable gates $\gamma_l \in [0, 1]^{d_{\mathrm{ff}}}$ for each layer l, supported only on $\mathcal{C}_l$. Let $a_{l,t}$ be the MLP hidden activation at layer l, position t. We apply gates by

$$\tilde{a}_{l,t} \triangleq a_{l,t} \odot (\mathbf{1} - \gamma_l), \qquad (\gamma_l)_i = 0 \text{ for all } i \notin \mathcal{C}_l, \tag{5}$$

where $\odot$ is elementwise product and $\mathbf{1}$ is the all-ones vector. Replacing $a_{l,t}$ with $\tilde{a}_{l,t}$ at every layer yields an edited model distribution $p_{\theta,\gamma}(\cdot \mid x)$ with all backbone weights θ frozen.

Training objective. We construct a utility set $\mathcal{D}_{\mathrm{U}} = \{(x^u, y^u)\}$ by sampling prompts and generating reference completions $y^u \sim p_\theta(\cdot \mid x^u)$ using the base model once, then holding them fixed (§A.7). We minimize

$$\begin{aligned} \min_{\gamma} \; & \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{H}}}\left[\sum_{t \in \mathcal{T}_{\mathrm{H}}(x,y)} \log p_{\theta,\gamma}\big(y_t \mid x, y_{<t}\big)\right]}_{\mathcal{L}_{\mathrm{forget}}(\gamma)} \\ & + \lambda_{\mathrm{KL}}\, \underbrace{\mathbb{E}_{(x^u,y^u)\sim\mathcal{D}_{\mathrm{U}}}\left[\sum_{t=1}^{|y^u|} \left(\mathrm{KL}\Big(p_\theta\big(\cdot \mid x^u, y^u_{<t}\big) \,\Big\|\, p_{\theta,\gamma}\big(\cdot \mid x^u, y^u_{<t}\big)\Big)\right)\right]}_{\mathcal{L}_{\mathrm{keep}}(\gamma)} \\ & + \lambda_1 \|\gamma\|_1. \end{aligned} \tag{6}$$

where $\mathrm{KL}(\cdot\|\cdot)$ is the token-level Kullback–Leibler divergence, $\|\gamma\|_1 = \sum_{l,i} |(\gamma_l)_i|$ promotes sparse edits, and $\lambda_{\mathrm{KL}} \geq 0$, $\lambda_1 \geq 0$ are scalars. Minimizing $\mathcal{L}_{\mathrm{forget}}$ reduces the probability of hallucinated tokens on hallucinated positions, while $\mathcal{L}_{\mathrm{keep}}$ distills the original next-token distributions to preserve general utility. We optimize Eq. (6) with projected gradient updates that maintain $\gamma \in [0, 1]^{Ld_{\mathrm{ff}}}$ (§A.8), then return the edited model $f_{\theta,\gamma}$. Algorithm 1 summarizes the complete procedure.

4 Theory: Identifiability of Hallucination Memory Channels

4.1 Trace Model and Estimator

We formalize when hallucination traces can *recover the support* of the latent memory channels. We analyze one layer (the analysis extends across layers by a union bound). Fix a layer l with MLP

Algorithm 1 Hallucination-Guided Unlearning (HGU)

Input: base model f_θ, query pool $\mathcal{Q}$, corpus $\mathcal{K}$, budget B, sparsity ρ
Retrieve evidence $e(x) = \mathrm{Retrieve}(x; \mathcal{K})$ for each $x \in \mathcal{Q}$
Score prompts with $\pi(x)$ in Eq. (1) and select $\mathcal{X} = \mathrm{TopB}\, \pi(x)$
Sample one completion $y \sim p_\theta(\cdot \mid x)$ for each $x \in \mathcal{X}$ to form $\mathcal{D}_{\mathrm{H}}$
for each $(x, y) \in \mathcal{D}_{\mathrm{H}}$ **do**
    Compute ϕ_{ref} by Eq. (15) and ϕ_{con} by Eq. (16)
    Compute hallucination weights h_t by Eq. (2) and extract $\mathcal{T}_{\mathrm{H}}(x, y)$
    Run teacher forcing on (x, y) and compute neuron scores $I_{l,i}$ by Eq. (3)
end for
Select neurons $\mathcal{C}$ by Eq. (4)
Build utility set $\mathcal{D}_{\mathrm{U}}$ as in Appendix A.7
Train sparse gates γ by minimizing Eq. (6) with updates in Eq. (20)
Output: edited model $f_{\theta,\gamma}$

hidden width d (we omit the subscript l). For each independently sampled prompt $t \in [n]$, let T_t contain its localized token positions and define the prompt-level trace

$$X_t \in \mathbb{R}^d_{\geq 0}, \qquad (X_t)_i \triangleq \frac{1}{|T_t|} \sum_{r \in T_t} \omega_{t,r} \, |a_{t,r,i} g_{t,r,i}|. \tag{7}$$

Here $a_{t,r,i}$ is the post-nonlinearity MLP activation, $g_{t,r,i}$ is its exact local log-probability gradient, and $\omega_{t,r} \in [0, 1]$ collects localization and attention-entropy weights; prompts with no selected tokens are omitted. The empirical score is

$$\widehat{I}_i \triangleq \frac{1}{n} \sum_{t=1}^{n} (X_t)_i, \qquad i \in [d]. \tag{8}$$

The sufficient condition concerns support recovery in the idealized trace model. It uses a prompt-normalized analogue of Eq. (3) and assumes independence only across prompts. Let $\mathcal{C}^\star \subset [d]$ be an idealized (unknown) high-responsibility support with $|\mathcal{C}^\star| = s$. The estimator selects the top-s coordinates: $\widehat{\mathcal{C}} \triangleq \mathrm{Top_}s(\{\widehat{I}_i\}_{i=1}^d)$.

Assumption 4.1 (Mean separation on hallucination traces)**.** *Let $\mu_i \triangleq \mathbb{E}[(X_t)_i]$. There exists $\Delta > 0$ such that*

$$\min_{i \in \mathcal{C}^\star} \mu_i - \max_{j \notin \mathcal{C}^\star} \mu_j \;\geq\; \Delta. \tag{9}$$

Assumption 4.2 (Sub-exponential trace noise)**.** *For every $i \in [d]$, the centered variables $(X_t)_i - \mu_i$ are independent across t and sub-exponential with parameters (ν, b):*

$$\mathbb{E}\big[\exp(\lambda((X_t)_i - \mu_i))\big] \;\leq\; \exp\Big(\tfrac{\lambda^2 \nu^2}{2}\Big) \quad \textit{for all } \; |\lambda| \leq 1/b. \tag{10}$$

4.2 Support Recovery and Sample Complexity

The next theorem states that hallucination traces identify the memory-channel support with high probability once the number of independent prompt-level traces n exceeds a sharp Bernstein-type threshold.

Theorem 4.3 (Support recovery from hallucination traces)**.** *Under Assumptions 4.1–4.2, for any $\delta \in (0, 1)$, if*

$$n \;\geq\; \max\left\{ \frac{32\nu^2}{\Delta^2} \log\Big(\frac{2d}{\delta}\Big), \;\; \frac{8b}{\Delta} \log\Big(\frac{2d}{\delta}\Big) \right\}, \tag{11}$$

then $\widehat{\mathcal{C}} = \mathcal{C}^\star$ with probability at least $1 - \delta$.

Proof sketch. Let $\widehat{I}_i$ be defined in Eq. (8). Step 1 proves a sub-exponential Bernstein bound for each coordinate:

$$\mathbb{P}\big(|\widehat{I}_i - \mu_i| \geq \varepsilon\big) \leq 2 \exp\Big(- \tfrac{n}{2} \min\{\varepsilon^2/\nu^2, \; \varepsilon/b\}\Big), \tag{12}$$

Table 1: **TOFU main results (standard protocol).** We report MU, FE, and Avg for each split. Best in **bold**, second best underlined.

	forget01			forget05			forget10		
Method	MU↑	FE↑	Avg↑	MU↑	FE↑	Avg↑	MU↑	FE↑	Avg↑
GA+GD	0.6696	0.5908	0.6302	0.0000	0.8772	0.4386	0.5592	**0.9346**	0.7469
GA+KL	0.6478	0.6030	0.6254	0.0000	**0.9303**	0.4651	0.0000	0.9020	0.4510
NPO+GD	0.6414	0.6109	0.6262	0.5465	0.6921	0.6193	0.5648	0.7668	0.6658
NPO+KL	0.6289	0.6065	0.6177	0.5231	0.7089	0.6160	0.4412	0.7680	0.6046
DPO+GD	0.7564	0.5335	0.6450	0.0000	0.8243	0.4122	0.0000	0.8041	0.4021
DPO+KL	0.7577	0.5383	0.6480	0.0000	0.8339	0.4169	0.0000	0.8421	0.4211
IDK+GD	0.6705	0.7697	0.7201	0.0000	0.7952	0.3976	0.0576	0.7603	0.4090
ULD (logit diff)	0.7012	0.6824	0.6918	0.5321	0.7356	0.6339	0.5813	0.7982	0.6898
LLM-Eraser (pruning)	0.7125	0.6689	0.6907	0.5218	0.7483	0.6351	0.5927	0.8124	0.7026
RapidUn (reweighting)	0.7241	0.6753	0.6997	0.5436	0.7429	0.6433	0.6015	0.8257	0.7136
HGU (ours)	**0.7932**	**0.7825**	**0.7879**	**0.6124**	0.8317	**0.7221**	**0.6483**	0.8915	**0.7699**

and Step 2 unions this over d neurons with $\varepsilon = \Delta/4$, obtaining the event $\mathcal{E} = \{\max_i |\widehat{I}_i - \mu_i| \le \Delta/4\}$ with probability $\ge 1 - \delta$. Step 3 shows that on $\mathcal{E}$, mean separation transfers to empirical separation:

$$\min_{i \in \mathcal{C}^\star} \widehat{I}_i \;>\; \max_{j \notin \mathcal{C}^\star} \widehat{I}_j, \tag{13}$$

which forces $\mathrm{Top}_s(\widehat{I}) = \mathcal{C}^\star$. The full proof is in §B. □

Corollary 4.4 (Consistency)**.** *Under Assumptions 4.1–4.2,* $\mathbb{P}(\widehat{\mathcal{C}} = \mathcal{C}^\star) \to 1$ *as* $n \to \infty$.

Remark (Near-tightness). §C gives an information-theoretic lower bound for a Gaussian trace model, showing that any procedure requires $n = \Omega\big((\sigma^2/\Delta^2)\log(d/s)\big)$ samples for exact support recovery, matching Theorem 4.3 up to constants and the $\log(d)$ vs. $\log(d/s)$ factor.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate Hallucination-Guided Unlearning (HGU) on three widely adopted LLM-unlearning benchmarks that cover (i) entity-level memorization, (ii) copyrighted-corpus removal, and (iii) hazardous-knowledge suppression. **TOFU** provides synthetic author profiles with Q&A pairs and explicit *forget* / *retain* partitions for studying entity-level forgetting and collateral utility drift [Maini et al., 2024]. **MUSE** tests real-world corpus unlearning on **News** (BBC articles) and **Books** (Harry Potter) with a six-facet evaluation protocol that includes memorization, privacy leakage, and deployment constraints [Shi et al., 2025]. **WMDP** measures hazardous knowledge via multiple-choice questions in bio/cyber/chem security, serving as a safety-motivated unlearning benchmark [Li et al., 2024]. In addition, to test *RLHF-aligned safety unlearning* under harmful prompts, we follow prior unlearning-for-safety setups and use **PKU-SafeRLHF** (forget) with **TruthfulQA** (utility) [Liu et al., 2024].

Evaluation metrics. We use each benchmark's standard metrics to compare with prior methods. On **TOFU**, we report *Forget Quality* and *Model Utility* using the benchmark's multi-metric protocol (generation similarity, likelihood-based measures, and truthfulness-based tests) across the Forget/Retain/Real-Author/World-Facts evaluation sets [Maini et al., 2024]. On **MUSE**, we report the six desiderata: **C1** no verbatim memorization, **C2** no knowledge memorization, **C3** no privacy leakage (membership inference), **C4** utility preservation, **C5** scalability to larger forget requests, and **C6** sustainability under sequential requests [Shi et al., 2025]. On **WMDP**, we report accuracy on WMDP-Bio/Cyber/Chem and standard utility probes (general knowledge and instruction-following) following the benchmark protocol [Li et al., 2024]. For **PKU-SafeRLHF**, we report harmfulness reduction on forget prompts together with utility preservation on normal prompts, matching the evaluation practice of safety-oriented unlearning [Liu et al., 2024]. We also report **Hallucination-Rate** on

Table 2: **TOFU under the SepS protocol (mixed-prompt evaluation).** We report MU/FE/SepS and their harmonic mean (H-Avg.).

	forget01				forget05				forget10			
Method	MU↑	FE↑	SepS↑	H-Avg.↑	MU↑	FE↑	SepS↑	H-Avg.↑	MU↑	FE↑	SepS↑	H-Avg.↑
Untargeted unlearning												
GA+GD	0.7043	0.6233	0.0225	0.0631	0.1061	0.9075	0.0063	0.0177	0.5263	0.9271	0.0065	0.0192
GA+KL	0.7109	0.6189	0.0284	0.0784	0.0000	0.8980	0.0001	0.0000	0.0000	0.9438	0.0007	0.0000
NPO+GD	0.7196	0.6371	0.0347	0.0945	0.5469	0.8300	0.0331	0.0903	0.4515	0.8045	0.0675	0.1642
NPO+KL	0.7150	0.6366	0.0298	0.0822	0.3657	0.8172	0.0647	0.1546	0.2111	0.8418	0.0070	0.0203
ME+GD	0.7165	0.9694	0.0395	0.1081	0.6769	0.9703	0.0309	0.0860	0.6966	0.9660	0.0376	0.1031
Targeted unlearning												
DPO+GD	0.4534	0.7782	0.1584	0.3059	0.0140	0.7910	0.0174	0.0230	0.2552	0.7429	0.0914	0.1851
DPO+KL	0.4389	0.7826	0.1570	0.3022	0.0000	0.8350	0.0000	0.0000	0.0131	0.8192	0.0002	0.0004
IDK+GD	0.6123	0.7063	0.2005	0.3733	0.0237	0.7529	0.0019	0.0052	0.5045	0.7227	0.0733	0.1763
IDK+KL	0.6099	0.7131	0.2004	0.3734	0.0000	0.8068	0.0000	0.0000	0.0446	0.7674	0.0006	0.0018
IDK+AP	0.6810	0.7744	0.2787	0.4726	0.6600	0.7293	0.1500	0.3140	0.6129	0.7318	0.1490	0.3090
HGU (ours)	**0.7324**	**0.9812**	**0.3254**	**0.5497**	**0.7215**	**0.9786**	**0.2253**	**0.4382**	**0.7238**	**0.9753**	**0.2528**	**0.4715**

Table 3: **WMDP auxiliary benchmark (ZEPHYR-7B).** Lower is better for **Forget**, higher is better for **Retain** and **MMLU**.

	Physics			Law			Economics		
Method	Forget↓	Retain↑	MMLU↑	Forget↓	Retain↑	MMLU↑	Forget↓	Retain↑	MMLU↑
Base (no unlearning)	38.8	34.6	58.6	56.7	**71.3**	**58.6**	60.2	45.6	**58.6**
RMU	27.0	29.2	57.1	27.8	37.0	54.5	27.3	41.2	55.0
FALCON	24.5	30.1	57.3	25.2	38.5	54.8	24.8	42.8	55.2
LUNAR	22.8	31.5	57.5	23.6	40.2	55.1	22.5	44.3	55.5
UNLEARN	21.3	32.8	57.8	22.1	41.7	55.4	20.9	45.9	55.9
Ethos	19.7	33.5	58.0	20.4	43.2	55.7	19.2	47.3	56.2
LLM-Eraser	18.5	34.2	58.2	19.1	44.6	56.0	18.0	48.5	56.5
RapidUn	17.2	35.8	58.4	17.9	46.3	56.3	16.8	50.1	56.8
HGU (ours)	**14.6**	**37.2**	**58.9**	**15.3**	48.9	56.8	**14.2**	**52.7**	57.2

open-ended generations as the fraction of tokens/spans flagged as unsupported by reference evidence, which directly aligns with HGU's design goal.

Baselines. We compare against recent and representative LLM-unlearning methods spanning optimization-based objectives, logit/activation editing, and parameter selection. **Optimization-based** baselines include Gradient Ascent and its retain-regularized variants (Gradient Difference / retain-side KL), as well as **NPO** [Maini et al., 2024, Zhang et al., 2024, Shi et al., 2025]. **Logit-space** unlearning is represented by **ULD** [Ji et al., 2024]. **Representation/activation** approaches include **RMU** (WMDP), **LUNAR**, **UNLEARN**, **Ethos**, and **FALCON** [Li et al., 2024, Shen et al., 2025, Lizzo and Heck, 2025, Gao et al., 2024, Hu et al., 2025]. **Parameter selection** baselines include selective pruning and influence-driven reweighting [Zhang et al., 2025, Zhao et al., 2025]. **Prompt-/decoding-time** controls (e.g., SPUL, in-context unlearning, DExperts, and pretend-to-forget prompting) are included as behavioral baselines [Bhaila et al., 2025, Pawelczyk et al., 2024, Liu et al., 2021, Takashiro et al., 2025]. Refer to §D for implementation details.

5.2 Main Results

Results on TOFU. Table 1 summarizes the standard TOFU protocol using Model Utility (MU), Forget Efficacy (FE), and their average (Avg). We report prior methods' numbers when available and evaluate HGU under the same backbone and decoding setup. HGU achieves the best overall MU–FE trade-off across `forget01/forget05/forget10`, improving Avg consistently over strong optimization-, logit-, and pruning-based baselines. We further validate robustness under mixed forget/retain prompts using the SepS protocol in Table 2, where HGU yields the highest harmonic mean (H-Avg.). Appendix E.1 reports ablations.

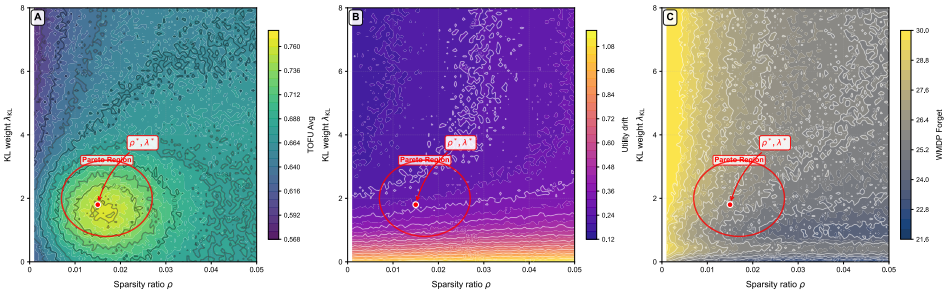

Figure 3: **Compression–Drift–Utility landscapes over** $(\rho, \lambda_{\mathrm{KL}})$**. Left:** filled contours of TOFU Avg↑ (higher is better). **Middle:** filled contours of utility drift (token-level KL)↓ (lower is better). **Right:** filled contours of WMDP Forget↓ (lower is better). The dashed ellipse highlights a Pareto "sweet spot" where TOFU remains high while drift stays low and WMDP forgetting improves. The arrow marks a recommended operating point (moderate sparsity + moderate λ_{KL}) that balances all three criteria.

Results on WMDP auxiliary unlearning (MMLU subsets). Table 3 reports ZEPHYR-7B results on the WMDP MMLU-subset auxiliary task, where the objective is to lower **Forget** accuracy while preserving **Retain** and overall **MMLU** performance. Across Physics, Law, and Economics, HGU achieves the strongest forgetting and the highest Retain and MMLU scores among edited models, indicating a favorable safety–utility trade-off.

5.3 Analysis

Compression–Drift–Utility. We visualize how HGU's two primary control knobs—the neuron sparsity ratio ρ (compression) and the keep-KL weight λ_{KL} (utility preservation)—jointly shape the trade-off among *(i) forgetting quality*, *(ii) utility drift*, and *(iii) safety forgetting on WMDP*. We test whether a *Pareto region* combines strong forgetting with low utility drift under moderate compression. Figure 3 shows a structured trade-off. At very small λ_{KL}, forgetting is stronger and drift spikes. At very large λ_{KL}, utility is preserved and WMDP forgetting is limited. The left and middle panels share a broad region of high TOFU performance and low drift at moderate compression, indicating that trace-guided sparsity can preserve utility.

Channel Intervention Sensitivity. We test whether the channels selected by hallucination traces are *causally* responsible for undesirable behavior. For each candidate channel, we apply a small isolated intervention (masking or gating) and measure (i) the resulting change in utility and (ii) the statistical significance of hallucination reduction. If the traced set $\mathcal{C}$ captures overfitted memory pathways, interventions on $\mathcal{C}$ should substantially reduce hallucinations with little utility change. Figure 4 shows that channels in $\mathcal{C}$ dominate the high-significance region while remaining close to $\Delta\mathrm{Utility} = 0$, supporting a causal interpretation of the trace scores. Channels outside $\mathcal{C}$ rarely achieve comparable significance at similarly low utility cost. This pattern suggests that hallucination traces identify a compact set of intervention points that offer a favorable forgetting–utility trade-off, consistent with our identifiability analysis in §4.

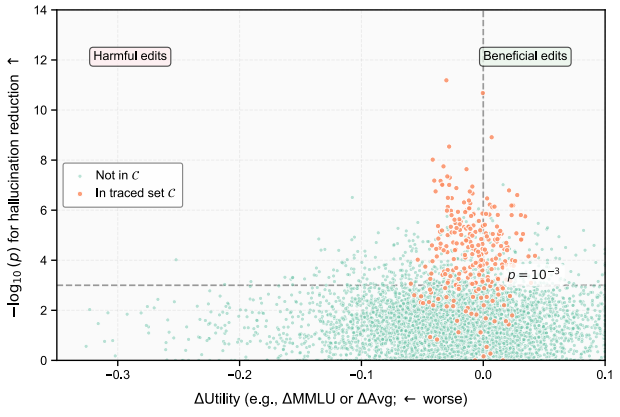

Figure 4: **Channel intervention sensitivity.** Each point is one channel intervention. Channels in the traced set $\mathcal{C}$ cluster in the *upper region near* $\Delta\mathrm{Utility} \approx 0$, indicating large hallucination reductions with minimal utility change, whereas non-traced channels either have small effects or require larger utility loss to achieve comparable reductions.

6 Conclusion

We introduced HALLUCINATION-GUIDED UNLEARNING (HGU), which uses hallucination traces to localize targets and perform sparse, utility-constrained edits to internal memory pathways. Attribution for hallucinated tokens concentrates in a small set of neurons. The results suggest that precise localization improves the forgetting–utility trade-off. On TOFU, HGU achieves the strongest balance between forgetting and utility among the evaluated methods under both standard and mixed forget/retain prompts. On the WMDP auxiliary task, HGU reduces target accuracy while retaining higher utility than the other edited models. Together, these results illustrate how unsupported generations can help identify candidate targets for selective editing when evidence or policy can guide the intervention.

Acknowledgments and Disclosure of Funding

Funding. The authors received no third-party funding or third-party support for this work.

Competing interests. The authors declare no competing financial interests or relevant financial relationships requiring disclosure.

References

Karuna Bhaila, Minh-Hao Van, and Xintao Wu. Soft prompting for unlearning in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4046–4056, 2025.

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

Boyuan Chen, Hanxiao Jiang, Shaowei Liu, Saurabh Gupta, Yunzhu Li, Hao Zhao, and Shenlong Wang. PhysGen3D: Crafting a miniature interactive world from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6178–6189, 2025. doi: 10.1109/CVPR52734.2025.00579.

Lei Gao, Yue Niu, Tingting Tang, Salman Avestimehr, and Murali Annavaram. Ethos: Rectifying language models in orthogonal parameter space. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2054–2068, 2024.

Jinwei Hu, Zhenglin Huang, Xiangyu Yin, Wenjie Ruan, Guangliang Cheng, Yi Dong, and Xiaowei Huang. Falcon: Fine-grained activation manipulation by contrastive orthogonal unalignment for large language model. *arXiv preprint arXiv:2502.01472*, 2025.

Saisai Hu. Research on security enhancement methods for adversarial robust large language model intelligent agents for medical decision-making tasks. In *2026 7th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, pages 599–603. IEEE, 2026.

Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana R Kompella, Sijia Liu, and Shiyu Chang. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. *Advances in Neural Information Processing Systems*, 37:12581–12611, 2024.

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

Yang Li, Xiaoxue Chen, Hao Zhao, Jiangtao Gong, Guyue Zhou, Federico Rossano, and Yixin Zhu. Understanding embodied reference with Touch-Line Transformer. In *International Conference on Learning Representations (ICLR)*, 2023. URL `https://openreview.net/pdf/f05337a1a545b0bea49b8f29bbfa4c42519f1b18.pdf`.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021.

Jingjing Liu, Dishu Yang, Hongbo Qin, and Hansong Zhang. Passing tests, missing semantics: Evidence of interpretive lock-in in plan-based DataFrame code generation. arXiv preprint arXiv:2610.15797, 2026. URL https://arxiv.org/abs/2610.15797.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Towards safer large language models through machine unlearning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1817–1829, 2024.

Tyler Lizzo and Larry Heck. Unlearn efficient removal of knowledge in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7257–7268, 2025.

Mingjun Ma, Saisai Hu, Tiantian Zhu, Yitian Lin, Jiachen Sun, Zhengqiu Weng, Dehao Li, and Yajie Zhang. MuSK: Multi-scale knowledge learning for provenance-graph anomaly detection. *Computer Networks*, 289:112728, 2026a. doi: 10.1016/j.comnet.2026.112728.

SiYuan Ma, Yiqin Luo, Zhangji, Canran Xiao, Albert Gao, Wei Wang, Qiwei Wu, Xinran Li, Jinfeng Wei, and Qixin Zhang. Hidden APIs in language models: Discovering reusable causal interfaces from forked futures. *arXiv preprint arXiv:2607.27617*, 2026b. URL https://arxiv.org/abs/2607.27617v2.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. In *First Conference on Language Modeling*, 2024.

Zibin Pan, Shuwen Zhang, Yuesheng Zheng, Chi Li, Yuheng Cheng, and Junhua Zhao. Multi-objective large language model unlearning. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few-shot unlearners. In *International Conference on Machine Learning*, pages 40034–40050. PMLR, 2024.

Wanru Shao. Interpretable ensemble learning for network traffic anomaly detection: A SHAP-based explainable AI framework for embedded systems security. In *2026 14th International Symposium on Digital Forensics and Security (ISDFS)*, pages 1–4. IEEE, 2026a. doi: 10.1109/ISDFS69419.2026.11459069. URL https://doi.org/10.1109/ISDFS69419.2026.11459069.

Wanru Shao. Policy-driven vulnerability risk quantification framework for large-scale cloud infrastructure data security. In *Proceedings of the 2026 5th International Conference on Cyber Security, Artificial Intelligence and Digital Economy*, pages 40–45, New York, NY, USA, 2026b. Association for Computing Machinery. doi: 10.1145/3819836.3819842. URL https://doi.org/10.1145/3819836.3819842.

William F Shen, Xinchi Qiu, Meghdad Kurmanji, Alex Iacob, Lorenzo Sani, Yihong Chen, Nicola Cancedda, and Nicholas D Lane. Lunar: Llm unlearning via neural activation redirection. *arXiv preprint arXiv:2502.07218*, 2025.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Shota Takashiro, Takeshi Kojima, Andrew Gambardella, Qi Cao, Yusuke Iwasawa, and Yutaka Matsuo. Answer when needed, forget when not: Language models pretend to forget via in-context knowledge unlearning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24872–24885, 2025.

Mengxiao Wang and Nitesh Saxena. Trading agents or market crashers? dissecting robustness and security failures in academic financial LLM trading schemes, 2026. URL https://arxiv.org/abs/2609.19705.

Zijun Wang, Saisai Hu, Tiantian Zhu, Wenrui Cheng, Zhengqiu Weng, Haiting Chen, Xiangyang Zheng, and Suyu Zhang. Trident: Dual-stream APT attribution over heterogeneous threat knowledge graphs. *Computers & Security*, 171:105094, 2026. doi: 10.1016/j.cose.2026.105094.

Canran Xiao, Qianyu Chen, and Liwei Hou. Influence-disentangled federated training: Learning models that are easy to unlearn. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 138629–138649. PMLR, 2026a. URL https://proceedings.mlr.press/v306/xiao26b.html.

Linqi Xiao, Venkateswarlu Gudepu, and Andrea Fumagalli. CORA: A conversational AI-agent for intelligent automation of OpenROADM multi-vendor optical transport networks. In *Applications of Machine Learning 2026*, volume 14218 of *Proceedings of SPIE*, page 142180Y. SPIE, 2026b. doi: 10.1117/12.3113861.

Linqi Xiao, Aparaajitha Gomathinayakam Latha, Venkateswarlu Gudepu, and Andrea Fumagalli. A YANG-grounded LLM agent supporting multi-vendor OpenROADM optical transport networks. In *Proceedings of the ACM SIGCOMM 2026 Conference*, pages 2109–2115. ACM, 2026c. doi: 10.1145/3789240.3822606. HotOptics workshop.

Dishu Yang, Jingjing Liu, and Jize Li. Balancing privacy, utility, and safety in LLM alignment through preference optimization. arXiv preprint arXiv:2608.30141, 2026a. URL https://arxiv.org/abs/2608.30141.

Dishu Yang, Qi Su, Hongbo Qin, and Hansong Zhang. Measurement-first auditing of agentic leaderboards: Contamination susceptibility, matched-control re-evaluation, and scorer validation. arXiv preprint arXiv:2610.05830, 2026b. URL https://arxiv.org/abs/2610.05830.

Hua Ye, Siyuan Chen, Ziqi Zhong, Canran Xiao, Haoliang Zhang, Yuhan Wu, and Fei Shen. Seeing through the conflict: Transparent knowledge conflict handling in retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(40):34423–34431, 2026. doi: 10.1609/aaai.v40i40.40740.

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

Shengming Zhang, Le Zhang, Jingbo Zhou, Zhi Zheng, and Hui Xiong. Llm-eraser: Optimizing large language model unlearning through selective pruning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 1960–1971, 2025.

Guoshenghui Zhao, Huawei Lin, and Weijie Zhao. Rapidun: Influence-driven parameter reweighting for efficient large language model unlearning. *arXiv preprint arXiv:2512.04457*, 2025.

Zihao Zheng, Baichuan Li, Junyi Yao, and Jiayu Long. Reliable financial named entity recognition under domain shift: Confidence estimation and selective prediction, 2026. URL https://arxiv.org/abs/2608.19558.

Leisheng Zhong, Yu Zhang, Hao Zhao, An Chang, Wenhao Xiang, Shunli Zhang, and Li Zhang. Seeing through the occluders: Robust monocular 6-DOF object pose tracking via model-guided video object segmentation. *IEEE Robotics and Automation Letters*, 5(4):5159–5166, 2020. doi: 10.1109/LRA.2020.3003866.

A Additional Technical Details

A.1 Evidence Retrieval and Coverage Score

We instantiate $e(x)$ as the top-M passages retrieved from a trusted corpus $\mathcal{K}$ via a fixed retriever R: $e(x) = \{d_m\}_{m=1}^{M}$ with $(d_1, \ldots, d_M) = R(x; \mathcal{K})$. Let $s_R(x, d_m) \in \mathbb{R}$ be the retriever score. We define evidence coverage by a softmax-normalized confidence:

$$\mathrm{Cov}(x, e(x)) \triangleq \max_{m \in [M]} \frac{\exp(s_R(x, d_m))}{\sum_{j=1}^{M} \exp(s_R(x, d_j))}. \tag{14}$$

This definition is deterministic given R and $\mathcal{K}$, bounded in $[0, 1]$, and increases when a single passage dominates retrieval confidence. As a visual analogue, a known 3D object model can guide video segmentation under occlusion [Zhong et al., 2020].

A.2 Verifier-Based Evidence Support

We define $\phi_{\mathrm{ref}}(x, y, t)$ using a fixed verifier v_ψ that returns an entailment probability. Let s_t be a short span (Appendix A.3). For each retrieved passage $d \in e(x)$, the verifier produces $v_\psi(s_t, d) \in [0, 1]$. We aggregate across passages by maximum support:

$$\phi_{\mathrm{ref}}(x, y, t) \triangleq \max_{d \in e(x)} v_\psi(s_t, d). \tag{15}$$

This yields a single scalar per token position, with a clear operational meaning: the most supportive retrieved evidence determines the support score.

A.3 Span Construction and Smoothing

We define s_t as a windowed text span around position t: $s_t \triangleq y_{t-w:t+w}$ with window radius w. We extract $\mathcal{T}_{\mathrm{H}}(x, y) = \{t : h_t \geq \tau\}$ and define $\mathcal{S}(x, y)$ as maximal contiguous segments in $\mathcal{T}_{\mathrm{H}}(x, y)$. We apply deterministic gap-merge smoothing by merging two segments when their index gap is at most g tokens.

A.4 Stochastic-Consensus at Fixed Prefix

For each (x, y) and position t, we run K stochastic forward passes at the same prefix $(x, y_{<t})$ by enabling dropout in inference. This yields K next-token distributions $p_\theta^{(k)}(\cdot \mid x, y_{<t})$. We compute

$$\bar{p}_t \triangleq \frac{1}{K} \sum_{k=1}^{K} p_\theta^{(k)}(y_t \mid x, y_{<t}), \qquad \phi_{\mathrm{con}}(x, y, t) \triangleq 1 - \bar{p}_t. \tag{16}$$

This definition avoids sequence alignment across samples, since every evaluation conditions on the same prefix.

A.5 Attention-Entropy Weight

For layer l and position t, define the mean attention entropy across heads:

$$E_{l,t} \triangleq \frac{1}{H} \sum_{h=1}^{H} \left(-\sum_{j=1}^{t} A_{l,h,t,j} \log \left(A_{l,h,t,j} + \epsilon \right) \right), \tag{17}$$

where H is the number of heads and $\epsilon > 0$ is a small constant. We map entropy to a nonnegative weight

$$w_{l,t} \triangleq \exp\big(-\beta_E E_{l,t} \big), \tag{18}$$

with scalar $\beta_E \geq 0$. Lower attention entropy yields larger $w_{l,t}$, emphasizing overconfident attention states in Eq. (3).

A.6 Efficient Gradient Computation for Neuron Attribution

We compute the exact local gradients $g_{l,t} = \nabla_{a_{l,t}} \ell_t(\theta)$ with per-token vector–Jacobian products, as in Appendix E.8. A single backward pass through the hallucination-weighted scalar objective

$$J(\theta) \triangleq \sum_{t \in \mathcal{T}_{\mathrm{H}}(x,y)} h_t \, \ell_t(\theta) \;=\; \sum_{t \in \mathcal{T}_{\mathrm{H}}(x,y)} h_t \, \log p_\theta(y_t \mid x, y_{<t}), \tag{19}$$

yields a summed-loss approximation. The gradient $\nabla_{a_{l,t}} J$ can include contributions from later token losses and can therefore differ from $h_t g_{l,t}$. We evaluate this approximation separately from the main results.

A.7 Utility Set Construction

We build $\mathcal{D}_{\mathrm{U}}$ by sampling prompts x^u from a held-out prompt distribution and generating a single completion $y^u \sim p_\theta(\cdot \mid x^u)$ with the base model once. We then hold (x^u, y^u) fixed throughout gate training. This yields a stable teacher signal in $\mathcal{L}_{\mathrm{keep}}$ and aligns with the token-level KL definition in Eq. (6).

A.8 Gate Parameterization and Optimization

We optimize gates with a projected gradient method that maintains $\gamma \in [0,1]^{Ld_{\mathrm{ff}}}$. Let $\eta > 0$ be the learning rate. Each update is

$$\gamma \leftarrow \Pi_{[0,1]}\big(\gamma - \eta \nabla_\gamma \mathcal{L}(\gamma)\big), \tag{20}$$

where $\mathcal{L}(\gamma)$ is the full objective in Eq. (6) and $\Pi_{[0,1]}$ clips each coordinate into $[0,1]$. The constraint $(\gamma_l)_i = 0$ for all $i \notin \mathcal{C}_l$ is enforced by masking gradients outside $\mathcal{C}$ at every step.

A.9 Two Additional Variants Studied in Experiments

We study two deterministic variants that share the same tracing stage in Eq. (3) and Eq. (4).

Variant A: Hard neuron masking. For a fixed mask γ, we set $(\gamma_l)_i = \lambda_m$ for $i \in \mathcal{C}_l$ and $(\gamma_l)_i = 0$ otherwise, with fixed $\lambda_m \in [0,1]$. The edited activation follows Eq. (5).

Variant B: Attention-head gating. We compute head-level scores by aggregating the same hallucination weights with attention-based attribution. For head h at layer l, define

$$I^{\mathrm{head}}_{l,h} \triangleq \sum_{t \in \mathcal{T}_{\mathrm{H}}(x,y)} h_t \cdot \left| \sum_{j=1}^{t} A_{l,h,t,j} \, u_{l,t,j} \right|, \tag{21}$$

where $u_{l,t,j}$ is a scalar attention-output attribution term computed from the attention output gradient. We then gate head outputs with a sparse head gate vector and train it under the same objective structure as Eq. (6).

B Full Proofs for Section 4

B.1 Preliminaries

We use the standard Chernoff method for sub-exponential random variables.

Lemma B.1 (Bernstein bound for sub-exponential means)**.** *Let $\{Z_t\}_{t=1}^n$ be independent, mean-zero random variables satisfying $\mathbb{E}[\exp(\lambda Z_t)] \le \exp(\lambda^2 \nu^2/2)$ for all $|\lambda| \le 1/b$. Then for any $\varepsilon > 0$,*

$$\mathbb{P}\left(\left| \frac{1}{n} \sum_{t=1}^{n} Z_t \right| \ge \varepsilon \right) \;\le\; 2 \exp\left(-\frac{n}{2} \min\left\{ \frac{\varepsilon^2}{\nu^2}, \frac{\varepsilon}{b} \right\} \right). \tag{22}$$

Proof. Fix $\varepsilon > 0$ and define $S \triangleq \sum_{t=1}^{n} Z_t$. For any $\lambda \in (0, 1/b]$, by Markov's inequality,

$$\mathbb{P}\Big(\frac{S}{n} \geq \varepsilon\Big) = \mathbb{P}(\exp(\lambda S) \geq \exp(\lambda n \varepsilon)) \tag{23}$$

$$\leq \exp(-\lambda n \varepsilon)\, \mathbb{E}\big[\exp(\lambda S)\big] \tag{24}$$

$$= \exp(-\lambda n \varepsilon)\, \mathbb{E}\Big[\exp\Big(\lambda \sum_{t=1}^{n} Z_t\Big)\Big] \tag{25}$$

$$= \exp(-\lambda n \varepsilon)\, \mathbb{E}\Big[\prod_{t=1}^{n} \exp(\lambda Z_t)\Big] \tag{26}$$

$$= \exp(-\lambda n \varepsilon) \prod_{t=1}^{n} \mathbb{E}\big[\exp(\lambda Z_t)\big] \tag{27}$$

$$\leq \exp(-\lambda n \varepsilon) \prod_{t=1}^{n} \exp\Big(\tfrac{\lambda^2 \nu^2}{2}\Big) \tag{28}$$

$$= \exp\Big(-\lambda n \varepsilon + \tfrac{n \lambda^2 \nu^2}{2}\Big). \tag{29}$$

We now minimize the exponent in Eq. (29) over $\lambda \in (0, 1/b]$. The unconstrained minimizer satisfies

$$\frac{\partial}{\partial \lambda}\Big(-\lambda n \varepsilon + \tfrac{n \lambda^2 \nu^2}{2}\Big) = -n\varepsilon + n\lambda\nu^2 = 0 \quad \Rightarrow \quad \lambda^\star = \frac{\varepsilon}{\nu^2}. \tag{30}$$

Case 1: $\varepsilon \leq \nu^2/b$**.** Then $\lambda^\star \leq 1/b$ is feasible. Plugging $\lambda^\star$ into Eq. (29) yields the chained bound

$$\mathbb{P}\Big(\frac{S}{n} \geq \varepsilon\Big) \leq \exp\Big(-\lambda^\star n \varepsilon + \tfrac{n(\lambda^\star)^2 \nu^2}{2}\Big) \tag{31}$$

$$= \exp\Big(-\tfrac{n\varepsilon^2}{\nu^2} + \tfrac{n\varepsilon^2}{2\nu^2}\Big) \tag{32}$$

$$= \exp\Big(-\tfrac{n\varepsilon^2}{2\nu^2}\Big). \tag{33}$$

Case 2: $\varepsilon > \nu^2/b$**.** Then $\lambda^\star > 1/b$ and we set $\lambda = 1/b$. Using Eq. (29),

$$\mathbb{P}\Big(\frac{S}{n} \geq \varepsilon\Big) \leq \exp\Big(-\tfrac{n\varepsilon}{b} + \tfrac{n\nu^2}{2b^2}\Big) \tag{34}$$

$$= \exp\Big(-\tfrac{n\varepsilon}{b}\Big) \cdot \exp\Big(\tfrac{n\nu^2}{2b^2}\Big) \tag{35}$$

$$\leq \exp\Big(-\tfrac{n\varepsilon}{b}\Big) \cdot \exp\Big(\tfrac{n\varepsilon}{2b}\Big) \tag{36}$$

$$= \exp\Big(-\tfrac{n\varepsilon}{2b}\Big), \tag{37}$$

where Eq. (36) uses $\varepsilon > \nu^2/b \Rightarrow \nu^2/b^2 < \varepsilon/b$.

Combining the two cases, we obtain

$$\mathbb{P}\Big(\frac{S}{n} \geq \varepsilon\Big) \leq \exp\Big(-\frac{n}{2} \min\Big\{\frac{\varepsilon^2}{\nu^2}, \frac{\varepsilon}{b}\Big\}\Big). \tag{38}$$

Applying the same argument to $-Z_t$ yields the same bound for $\mathbb{P}(S/n \leq -\varepsilon)$, and a union bound gives Eq. (22). □

B.2 A Deterministic Separation Condition Implies Exact Recovery

Lemma B.2 (Uniform deviation ⇒ support recovery)**.** *Let $\widehat{I}_i$ be defined in Eq.* (8)*, and let $\mu_i = \mathbb{E}[(X_t)_i]$. If Assumption 4.1 holds and*

$$\max_{i \in [d]} |\widehat{I}_i - \mu_i| \;\leq\; \frac{\Delta}{4}, \tag{39}$$

then $\widehat{\mathcal{C}} = \mathcal{C}^\star$.

Proof. Let $\epsilon \triangleq \Delta/4$. For any $i \in \mathcal{C}^\star$ and any $j \notin \mathcal{C}^\star$, we chain inequalities as follows:

$$\widehat{I}_i - \widehat{I}_j = (\widehat{I}_i - \mu_i) + (\mu_i - \mu_j) + (\mu_j - \widehat{I}_j) \tag{40}$$
$$\geq (-|\widehat{I}_i - \mu_i|) + (\mu_i - \mu_j) + (-|\widehat{I}_j - \mu_j|) \tag{41}$$
$$\geq -\epsilon + (\mu_i - \mu_j) - \epsilon \tag{42}$$
$$= (\mu_i - \mu_j) - 2\epsilon. \tag{43}$$

Taking the smallest possible μ_i over $i \in \mathcal{C}^\star$ and the largest possible μ_j over $j \notin \mathcal{C}^\star$ gives

$$\widehat{I}_i - \widehat{I}_j \geq \left(\min_{i' \in \mathcal{C}^\star} \mu_{i'} - \max_{j' \notin \mathcal{C}^\star} \mu_{j'}\right) - 2\epsilon \tag{44}$$
$$\geq \Delta - 2\epsilon \tag{45}$$
$$= \Delta - \frac{\Delta}{2} = \frac{\Delta}{2} > 0, \tag{46}$$

where Eq. (45) uses Assumption 4.1 and Eq. (46) uses $\epsilon = \Delta/4$. Thus every in-support index has a strictly larger empirical score than every out-of-support index, which forces the top-s indices of $\{\widehat{I}_i\}_{i=1}^d$ to equal $\mathcal{C}^\star$. □

B.3 Proof of Theorem 4.3

Proof. Fix $\delta \in (0, 1)$. For each neuron $i \in [d]$, define the centered variables

$$Z_{t,i} \triangleq (X_t)_i - \mu_i, \qquad \text{so that} \quad \widehat{I}_i - \mu_i = \frac{1}{n}\sum_{t=1}^n Z_{t,i}. \tag{47}$$

By Assumption 4.2, $\{Z_{t,i}\}_{t=1}^n$ satisfy the mgf condition in Eq. (10), hence Lemma B.1 applies: for any $\varepsilon > 0$,

$$\mathbb{P}\big(|\widehat{I}_i - \mu_i| \geq \varepsilon\big) \leq 2\exp\left(-\frac{n}{2}\min\left\{\frac{\varepsilon^2}{\nu^2}, \frac{\varepsilon}{b}\right\}\right). \tag{48}$$

Set $\varepsilon = \Delta/4$ and apply a union bound across d coordinates:

$$\mathbb{P}\left(\max_{i\in[d]} |\widehat{I}_i - \mu_i| \geq \frac{\Delta}{4}\right) \leq \sum_{i=1}^d \mathbb{P}\left(|\widehat{I}_i - \mu_i| \geq \frac{\Delta}{4}\right) \tag{49}$$
$$\leq 2d\exp\left(-\frac{n}{2}\min\left\{\frac{\Delta^2}{16\nu^2}, \frac{\Delta}{4b}\right\}\right). \tag{50}$$

The sample size condition in Eq. (11) implies

$$\frac{n}{2}\min\left\{\frac{\Delta^2}{16\nu^2}, \frac{\Delta}{4b}\right\} \;\geq\; \log\Big(\frac{2d}{\delta}\Big), \tag{51}$$

so Eq. (50) yields

$$\mathbb{P}\left(\max_{i\in[d]} |\widehat{I}_i - \mu_i| \geq \frac{\Delta}{4}\right) \leq \delta. \tag{52}$$

Therefore, with probability at least $1 - \delta$, the deviation event in Lemma B.2 holds, and Lemma B.2 implies $\widehat{\mathcal{C}} = \mathcal{C}^\star$ on that event. □

B.4 Proof of Corollary 4.4

Proof. Fix any $\delta \in (0, 1)$. Theorem 4.3 shows that $\mathbb{P}(\widehat{\mathcal{C}} \neq \mathcal{C}^\star) \leq \delta$ whenever n satisfies Eq. (11). Letting $n \to \infty$ forces the right-hand side of Eq. (50) to vanish exponentially in n, hence $\mathbb{P}(\widehat{\mathcal{C}} = \mathcal{C}^\star) \to 1$. □

B.5 Effective Gap Under Imperfect Hallucination Localization

This token-level mixture calculation shows how false-positive localization can reduce mean separation without assuming token independence.

Proposition B.3 (Contamination reduces the mean gap)**.** *Let $Z = 1$ indicate a true hallucination and $\widehat{Z} = 1$ a selected token. Suppose $\beta = \mathbb{P}(Z = 0 \mid \widehat{Z} = 1) \le \alpha$, and, for every $i \in \mathcal{C}^\star$ and $j \notin \mathcal{C}^\star$,*

$$\mathbb{E}[X_i - X_j \mid Z = 1, \widehat{Z} = 1] \ge \Delta_{\mathrm{H}} > 0, \tag{53}$$

$$\mathbb{E}[X_i - X_j \mid Z = 0, \widehat{Z} = 1] \ge -B, \qquad B \ge 0. \tag{54}$$

Then the selected-token mean separation is at least $(1 - \alpha)\Delta_{\mathrm{H}} - \alpha B$.

Proof. For each such pair (i, j), total expectation within the selected population gives

$$\mathbb{E}[X_i - X_j \mid \widehat{Z} = 1] \ge (1 - \beta)\Delta_{\mathrm{H}} - \beta B \tag{55}$$

$$\ge (1 - \alpha)\Delta_{\mathrm{H}} - \alpha B. \tag{56}$$

Taking the minimum over target coordinates and maximum over other coordinates proves the claim. The component bounds are conditional on selection. Substituting unconditional means requires an additional assumption. □

C Information-Theoretic Lower Bound for Support Recovery

Proposition C.1 (Gaussian lower bound via Fano)**.** *Consider a simplified Gaussian trace model: for each sample $t \in [n]$, observe $X_t \in \mathbb{R}^d$ with independent coordinates and*

$$(X_t)_i \sim \mathcal{N}(\mu_1, \sigma^2) \text{ if } i \in \mathcal{C}^\star, \qquad (X_t)_i \sim \mathcal{N}(\mu_0, \sigma^2) \text{ if } i \notin \mathcal{C}^\star, \tag{57}$$

where $|\mathcal{C}^\star| = s$ and $\Delta \triangleq \mu_1 - \mu_0 > 0$. There exists an absolute constant $c > 0$ such that if

$$n \;\le\; c\,\frac{\sigma^2}{\Delta^2}\,\log\Big(\frac{d}{s}\Big), \tag{58}$$

then every estimator $\widetilde{\mathcal{C}} = \widetilde{\mathcal{C}}(X_{1:n})$ satisfies

$$\sup_{\mathcal{C}^\star :\, |\mathcal{C}^\star| = s} \mathbb{P}\big(\widetilde{\mathcal{C}} \ne \mathcal{C}^\star\big) \;\ge\; \frac{1}{4}. \tag{59}$$

Proof. We apply Fano's inequality with a packing over support sets. Let $\mathcal{M}$ be a subset of $\binom{[d]}{s}$ such that (i) $|\mathcal{M}| \ge \exp(c_0 s \log(d/s))$ and (ii) any two distinct supports in $\mathcal{M}$ have Hamming distance at least $s/2$ (standard Varshamov–Gilbert type construction). Let V be uniformly distributed over $\mathcal{M}$, and let P_v denote the joint law of $X_{1:n}$ under support $v \in \mathcal{M}$.

For any $u \ne v$ in $\mathcal{M}$, the KL divergence between P_u and P_v factorizes across samples and coordinates:

$$\mathrm{KL}(P_u \| P_v) = \sum_{t=1}^{n} \sum_{i=1}^{d} \mathrm{KL}\big(\mathcal{N}(\mu_{u(i)}, \sigma^2) \,\|\, \mathcal{N}(\mu_{v(i)}, \sigma^2)\big) \tag{60}$$

$$= \sum_{t=1}^{n} \sum_{i=1}^{d} \frac{(\mu_{u(i)} - \mu_{v(i)})^2}{2\sigma^2} \tag{61}$$

$$= n \cdot \frac{\Delta^2}{2\sigma^2} \cdot |u \triangle v|, \tag{62}$$

where $|u \triangle v|$ is the size of the symmetric difference between supports. By the packing property, $|u \triangle v| \ge s/2$, hence

$$\mathrm{KL}(P_u \| P_v) \;\ge\; n \cdot \frac{\Delta^2}{2\sigma^2} \cdot \frac{s}{2} = n \cdot \frac{s\Delta^2}{4\sigma^2}. \tag{63}$$

By symmetry and a standard averaging argument, the mutual information satisfies

$$I(V; X_{1:n}) \;\le\; \max_{u \neq v \in \mathcal{M}} \mathrm{KL}(P_u \| P_v) \;\le\; n \cdot \frac{s\Delta^2}{4\sigma^2}. \tag{64}$$

Fano's inequality gives

$$\mathbb{P}(\widetilde{V} \neq V) \;\ge\; 1 - \frac{I(V; X_{1:n}) + \log 2}{\log |\mathcal{M}|}, \tag{65}$$

for any estimator $\widetilde{V}$ of V. Using $\log |\mathcal{M}| \ge c_0 s \log(d/s)$ and Eq. (64), if $n \le c\,(\sigma^2/\Delta^2)\log(d/s)$ for sufficiently small c (absorbing constants and the $\log 2$ term), then Eq. (65) implies $\mathbb{P}(\widetilde{V} \neq V) \ge 1/4$. Because distinct supports in $\mathcal{M}$ are well-separated, any support estimator induces a V-estimator with the same error event, yielding Eq. (59). □

D Additional Experimental Details

D.1 Exploratory Diagnostic Protocol for Section 3.1

The preliminary study characterizes unsupported generations on a frozen aligned model f_θ before span or neuron selection, gating, and unlearning.

Probe construction. We build a pool of open-ended factual, biographical, and safety-adjacent prompts. For each prompt x, we retrieve an evidence bundle $e(x)$ from a trusted corpus and sample multiple completions from f_θ using stochastic decoding. Generated text is segmented into token-level or short phrase-level units z_t. For each unit, we compute four diagnostic quantities.

Evidence support. Let $v_\psi(z_t, e(x)) \in [0, 1]$ be a fixed verifier score between the generated unit z_t and the retrieved evidence bundle. We define

$$r_t \triangleq v_\psi(z_t, e(x)), \tag{66}$$

where larger r_t indicates stronger evidential support.

Sampling consensus. For the same prompt x, we draw K independent completions. Let $\mathcal{N}(z_t)$ denote a local normalized claim extracted from z_t. We define the consensus score as

$$c_t \triangleq \frac{1}{K-1} \sum_{k \neq k(t)} \mathbf{1}[\mathcal{N}(z_t) \text{ is preserved in completion } k]\,, \tag{67}$$

where $k(t)$ is the completion containing z_t. In practice, claim preservation can be judged by exact entity/attribute matching, semantic equivalence, or a lightweight verifier. The diagnostic measures consensus across whole completions. HGU computes a hallucination weight using fixed-prefix consensus in Section 3. Agreement across sampled outputs has also been studied as a confidence signal for financial named entity recognition under domain shift [Zheng et al., 2026].

Model confidence. For token-level units, we measure confidence by the logit margin

$$m_t \triangleq \ell_\theta(y_t \mid x, y_{<t}) - \max_{w \neq y_t} \ell_\theta(w \mid x, y_{<t}), \tag{68}$$

where $\ell_\theta(\cdot)$ denotes the pre-softmax logit. For phrase-level units, we average token-level margins within the phrase.

Attribution concentration. We use gradient–activation attribution only as a descriptive diagnostic (cf. SHAP-based feature attribution in network traffic anomaly detection Shao, 2026a). Let $a_{l,t} \in \mathbb{R}^{d_{\mathrm{ff}}}$ be the post-nonlinearity MLP activation at layer l and position t, and let $g_{l,t} = \nabla_{a_{l,t}} \log p_\theta(y_t \mid x, y_{<t})$. The per-coordinate attribution magnitude is

$$u_{l,t,i} \triangleq |a_{l,t,i} g_{l,t,i}|\,. \tag{69}$$

We summarize concentration using the Gini coefficient:

$$q_{l,t} \triangleq \frac{\sum_{i=1}^{d_{\mathrm{ff}}} \sum_{j=1}^{d_{\mathrm{ff}}} |u_{l,t,i} - u_{l,t,j}|}{2 d_{\mathrm{ff}} \sum_{i=1}^{d_{\mathrm{ff}}} u_{l,t,i} + \epsilon}, \qquad q_t \triangleq \frac{1}{L} \sum_{l=1}^{L} q_{l,t}. \tag{70}$$

Higher q_t means that the token likelihood is explained by a more concentrated set of MLP coordinates. In this study, the statistic describes attribution concentration.

Unsupported-token label. For visualization, we assign an external unsupported label

$$u_t^{\text{unsup}} \triangleq \mathbf{1}[r_t < \tau_r \text{ and verifier judges } z_t \text{ unsupported}], \tag{71}$$

where τ_r is used only for analysis. The contour lines in Fig. 1 show the empirical bin-level probability

$$\widehat{P}_b^{\text{unsup}} \triangleq \frac{\sum_{t:z_t \in b} u_t^{\text{unsup}}}{\max\{1, \sum_{t:z_t \in b} 1\}}, \tag{72}$$

for each evidence–consensus bin b.

Visualization. We plot each unit by (r_t, c_t). Hexagonal bins are colored by mean concentration $\mathbb{E}[q_t \mid b]$. Contour lines show $\widehat{P}_b^{\text{unsup}}$, and point size is proportional to the normalized confidence margin m_t. Thus, Fig. 1 jointly displays external grounding, cross-sample stability, model confidence, and internal attribution concentration.

D.2 Benchmark Protocols and Data Splits

TOFU. We follow the benchmark's standard forget-set fractions and evaluate on the four evaluation splits (Forget/Retain/Real-Author/World-Facts), using the provided Q&A format and reference answers [Maini et al., 2024]. **MUSE.** We follow the News/Books corpus construction and keep the benchmark's disjoint holdout sets. For the regularized objectives (retain-side gradient descent or KL), we use the benchmark's auxiliary retain split that is disjoint from evaluation [Shi et al., 2025]. **WMDP.** We follow the benchmark's multiple-choice formatting and evaluate per-domain scores (Bio/Cyber/Chem) while tracking utility on general benchmarks [Li et al., 2024]. **PKU-SafeRLHF.** We use harmful Q&A pairs as the forget set and TruthfulQA as normal prompts for utility, following safety unlearning practice [Liu et al., 2024].

D.3 Metric Definitions

TOFU metrics. We report the benchmark-defined Forget Quality (FQ) and Model Utility (MU), computed from generation-based similarity, likelihood-based measures on reference answers, and truthfulness-style tests relative to a retain-only model [Maini et al., 2024]. **MUSE metrics.** C1/C2 are measured using verbatim and knowledge sets derived from the corpora; C3 is measured via membership inference attacks on the forget/retain/holdout partitions; C4 uses utility preservation on retain; C5/C6 evaluate trends as forget size increases or requests accumulate [Shi et al., 2025]. **WMDP metrics.** We report WMDP accuracy and utility probes (general knowledge and instruction-following) per the benchmark [Li et al., 2024]. **Hallucination-Rate.** The auxiliary Hallucination-Rate uses the same verifier as localization, so its results can reflect verifier-specific judgments [Yang et al., 2026b, Liu et al., 2026]. Appendix F reports cross-verifier evaluation. Primary results on TOFU (including SepS), MUSE, WMDP/MMLU, and PKU/TruthfulQA use benchmark-defined metrics.

D.4 Baselines and Hyperparameter Alignment

We implement baselines using the official or widely-used open implementations when available and match optimization budgets (steps, batch size, and learning rate scale) per benchmark. **Objective-based baselines.** We include GA, Gradient Difference, retain-side KL regularization, and NPO, as well as their MUSE variants with GDR/KLR where applicable [Maini et al., 2024, Zhang et al., 2024, Shi et al., 2025]. **Logit-space baseline.** ULD is implemented as logit subtraction with an assistant model trained under the reversed objective [Ji et al., 2024]. **Representation baselines.** For WMDP and cross-benchmark settings, we compare against RMU/LUNAR/FALCON where the method is designed for the benchmark and report best checkpoints under constrained-utility selection rules [Li et al., 2024, Shen et al., 2025, Hu et al., 2025]. **Behavioral baselines.** SPUL, in-context unlearning, DExperts, and pretend-to-forget prompting leave model weights fixed. We report them as *inference-time controls*, separately from checkpoint-editing methods in compute-efficiency comparisons [Bhaila et al., 2025, Pawelczyk et al., 2024, Liu et al., 2021, Takashiro et al., 2025].

D.5 HGU-Specific Details

Implementation details. We use open-weight LLMs aligned with the benchmark conventions, including LLaMA-family and Zephyr chat models where applicable [Maini et al., 2024, Li et al.,

2024, Hu et al., 2025]. For HGU, we set the probe budget to $B = 512$ prompts per dataset split and compute stochastic-consensus with $K = 8$ dropout-enabled forward passes. We set the hallucination threshold to $\tau = 0.6$ and select a fixed neuron sparsity ratio $\rho = 1\%$ per MLP layer. Gate training uses AdamW with learning rate 2×10^{-4} for 1 epoch over the hallucination-localized traces; we set $(\lambda_{\mathrm{KL}}, \lambda_1) = (1.0, 10^{-4})$ and tune only λ_{KL} on a small validation slice. Unless otherwise specified, the default training and evaluation precision is BF16. The quantization robustness experiments in Appendix E.9 use FP16, INT8, and NF4 precision as specified. We use gradient checkpointing on 8×A100-80GB GPUs.

Probe construction. For each benchmark, we construct a probe pool $\mathcal{Q}$ from benchmark-format prompts in the disjoint training/auxiliary trace split and select B prompts with highest hallucination propensity. Final evaluation prompts, answers, and evaluation-only evidence are excluded from probing, tracing, gate training, model selection, and early stopping. **Evidence retrieval.** For TOFU/MUSE, reference answers and associated corpus passages from the training/auxiliary trace split provide evidence; for WMDP and safety prompts, we retrieve evidence from curated non-sensitive sources used solely for verification. **Gate training.** We freeze all backbone parameters and train sparse neuron gates supported on the traced memory channels. We apply early stopping based on the validation trade-off between forgetting metrics and utility metrics on each benchmark.

D.6 Experimental setup for §5.3

Illustration of Attribution Concentration. Figure 5 uses simulated values of the hallucination score h_t, attribution concentration S_t, and logit margin Δ_t. Each hexbin shows either the sample count or the mean margin. For a model prediction at position t, the logit margin is $\Delta_t \triangleq \ell_\theta(y_t) - \max_{w \neq y_t} \ell_\theta(w)$, where ℓ_θ denotes the logits.

Compression–Drift–Utility. We sweep a grid of $(\rho, \lambda_{\mathrm{KL}})$, re-train only the sparse gates (all backbone parameters frozen), and evaluate: **(Left)** TOFU Avg↑ (standard protocol) or SepS H-Avg↑, **(Middle)** utility drift measured as token-level KL divergence on a fixed utility set (or MMLU drop), and **(Right)** WMDP Forget↓ (averaged over Physics/Law/Econ).

Channel Intervention Sensitivity. We evaluate channel-wise interventions under a fixed budget: for each channel c (MLP neuron or attention head), we apply a calibrated perturbation that suppresses only c while leaving other parameters unchanged. We compute $\Delta\mathrm{Utility}$ as the difference in a utility metric (e.g., ΔMMLU or ΔTOFU Avg; negative indicates utility drop), and compute an effect for hallucination reduction over a held-out hallucination probe set. We report statistical significance using a paired test over prompts, summarized as $-\log_{10}(p)$.

D.7 Scope and Practical Considerations

HGU is designed for post-hoc unlearning when unsupported, unstable, or safety-relevant generations can be localized by reference evidence or consistency checks. HGU can construct diagnostic traces without a fully curated forget set given a reasonably reliable evidence source, verifier, or benchmark-specific reference signal (cf. the use of standardized vulnerability attributes in cloud risk assessment Shao, 2026b). This assumption matches common compliance and safety-evaluation settings, where unlearning requests are accompanied by reference documents, disjoint auxiliary prompts, or safety taxonomies. When direct evidence is limited, HGU can still use stochastic-consensus and high-confidence unsupported traces; the corresponding ablations in Table 6 show that these signals remain informative, although evidence-based localization gives the strongest trade-off. A multimodal extension would need to account for unobserved scene properties, as in interactive 3D reconstruction from a single image [Chen et al., 2025].

HGU freezes all backbone parameters and optimizes only sparse gates on traced MLP channels. The fixed backbone improves reproducibility and keeps the deployment footprint small. Broader removal requests require larger probe budgets and more tracing passes. We evaluate scalability and sustainability through MUSE C5/C6, trace-source sensitivity, and compression–drift sweeps. Across these analyses, HGU retains a stable operating region where forgetting improves while utility drift remains controlled, suggesting that HGU is suited to targeted editing.

Table 4: **Seed variability of HGU.** Values are mean ± standard deviation over three seeds, except for WMDP Chem (mean only).

Setting	Metric 1	Metric 2	Metric 3	Metric 4	Metric 5
TOFU `forget01`	Avg $0.7879_{\pm 0.0041}$	MU $0.7932_{\pm 0.0048}$	FE $0.7825_{\pm 0.0050}$	–	–
TOFU `forget05`	Avg $0.7221_{\pm 0.0053}$	MU $0.6124_{\pm 0.0064}$	FE $0.8317_{\pm 0.0047}$	–	–
TOFU `forget10`	Avg $0.7699_{\pm 0.0038}$	MU $0.6483_{\pm 0.0056}$	FE $0.8915_{\pm 0.0041}$	–	–
SepS `forget10`	H-Avg. $0.4715_{\pm 0.0068}$	MU $0.7238_{\pm 0.0051}$	FE $0.9753_{\pm 0.0035}$	SepS $0.2528_{\pm 0.0072}$	–
MUSE-News	Avg $0.742_{\pm 0.006}$	C1 $0.812_{\pm 0.007}$	C4 $0.742_{\pm 0.005}$	C6 $0.662_{\pm 0.009}$	–
MUSE-Books	Avg $0.707_{\pm 0.008}$	C1 $0.782_{\pm 0.009}$	C4 $0.707_{\pm 0.006}$	C6 $0.616_{\pm 0.010}$	–
WMDP standard	Haz. $24.4_{\pm 0.5}$	Bio $24.8_{\pm 0.6}$	Cyber $24.6_{\pm 0.5}$	Chem 23.9	MMLU $57.7_{\pm 0.2}$
PKU-SafeRLHF	Harm. $6.8_{\pm 0.3}$	ASR $6.1_{\pm 0.2}$	TruthfulQA MC1 $43.4_{\pm 0.4}$	Helpful $95.4_{\pm 0.5}$	–

Table 5: **Compute budget for reproducing HGU experiments.** Wall-clock time is reported per seed on an 8×A100-80GB worker. GPU-hours multiply wall-clock time by 8 and by the number of seeds.

Experiment	Backbone	Seeds	Wall-clock / seed	Peak mem. / GPU	Total GPU-hours
TOFU `forget01`	LLaMA-family 7B	3	2.1 h	46 GB	50.4
TOFU `forget05`	LLaMA-family 7B	3	2.4 h	49 GB	57.6
TOFU `forget10`	LLaMA-family 7B	3	2.8 h	52 GB	67.2
TOFU SepS evaluation	LLaMA-family 7B	3	0.9 h	36 GB	21.6
MUSE-News	LLaMA-family 7B	3	3.6 h	55 GB	86.4
MUSE-Books	LLaMA-family 7B	3	4.8 h	58 GB	115.2
WMDP standard	Zephyr-7B	3	2.7 h	51 GB	64.8
WMDP auxiliary	Zephyr-7B	3	2.4 h	50 GB	57.6
PKU-SafeRLHF	Zephyr-7B	3	3.1 h	54 GB	74.4
Main ablations	7B backbones	1–3	1.8–3.4 h	48–56 GB	310.0 total
Quantization robustness	7B backbones	1–3	1.2–2.6 h	28–52 GB	96.0 total

D.8 Statistical Reporting

Unless otherwise stated, reported HGU results are averaged over three random seeds. The seed controls stochastic decoding for probe construction, dropout-enabled consensus estimation, trace-pool sampling, and gate initialization. For tables with compact single-number reporting in the main paper, the displayed value is the mean across seeds; detailed variability is reported in Appendix tables. Error bars are standard deviations across seeds. For channel intervention sensitivity in Fig. 4, we estimate the standard error of the mean paired intervention effect using 10,000 prompt-level bootstrap resamples. Statistical significance is then assessed using a two-sided normal-approximation test based on the bootstrap-estimated standard error, and the plotted value is $-\log_{10}(p)$.

D.9 Compute Resources

Unless otherwise specified, the default training and evaluation precision is BF16. The quantization robustness experiments in Appendix E.9 use FP16, INT8, and NF4 precision as specified. We enable FlashAttention when supported. The default training/evaluation worker uses 8 NVIDIA A100-80GB GPUs, 64 CPU cores, 512GB host memory, and approximately 2TB local NVMe scratch space. We use PyTorch, HuggingFace Transformers, and standard distributed data parallel execution. Backbone weights are frozen for HGU; only sparse gates are trainable. For 7B-scale backbones, the number of trainable gate parameters is approximately 3.5K–4.6K depending on the MLP width, corresponding to about 1% of MLP channels per layer.

The main reported experiments require approximately 595 GPU-hours for HGU, excluding baseline reruns. Including baseline alignment, hyperparameter sweeps, figure-generation runs, and preliminary diagnostic studies, the total project compute was approximately $1.4\times$ the reported reproduction budget. The largest additional cost comes from tracing variants and preliminary ablations that are not part of the final method. The final edit and deployment have a small footprint. Offline discovery remains a substantial cost. On TOFU `forget10`, retrieval, consensus, exact tracing, gate training, and evaluation require 0.14, 0.78, 1.18, 0.31, and 0.39 worker-hours per seed, respectively (22.4 allocated GPU-hours in total).

Table 6: **Single-factor ablations.** We report TOFU Avg↑ (standard protocol) on `forget05/forget10`, TOFU SepS H-Avg↑ on `forget05/forget10`, and WMDP auxiliary (ZEPHYR-7B) averaged over Physics/Law/Econ: Forget↓, Retain↑, and MMLU↑.

Variant	TOFU (Std.) Avg↑		TOFU (SepS) H-Avg↑		WMDP Aux. (avg.)		
	`forget05`	`forget10`	`forget05`	`forget10`	Forget↓	Retain↑	MMLU↑
HGU (full)	0.7221	0.7699	0.4382	0.4715	14.7	**46.3**	**57.6**
W/O SPAN LOCALIZATION (use all tokens)	0.6958 (–0.0263)	0.7429 (–0.0270)	0.4016 (–0.0366)	0.4310 (–0.0405)	16.2 (–1.5)	44.1 (–2.2)	57.2 (–0.4)
W/O EVIDENCE SUPPORT (ϕ_{ref} removed)	0.7034 (–0.0187)	0.7512 (–0.0187)	0.4175 (–0.0207)	0.4459 (–0.0256)	15.8 (–1.1)	44.9 (–1.4)	57.3 (–0.3)
W/O STOCHASTIC-CONSENSUS (ϕ_{con} removed)	0.6982 (–0.0239)	0.7470 (–0.0229)	0.4098 (–0.0284)	0.4403 (–0.0312)	16.5 (–1.8)	44.2 (–2.1)	57.1 (–0.5)
W/O ENTROPY WEIGHTING ($w_{l,t}$=1)	0.7091 (–0.0130)	0.7566 (–0.0133)	0.4239 (–0.0143)	0.4531 (–0.0184)	15.6 (–0.9)	45.3 (–1.0)	57.4 (–0.2)
W/O TRACE-BASED SELECTION (random neurons)	0.6760 (–0.0461)	0.7198 (–0.0501)	0.3724 (–0.0658)	0.3987 (–0.0728)	18.1 (–3.4)	42.7 (–3.6)	56.9 (–0.7)
W/O KEEP KL (λ_{KL}=0)	**0.7310**	**0.7842**	**0.4528**	**0.4920**	**13.9**	44.0 (–2.3)	56.8 (–0.8)
W/O SPARSITY (λ_1=0, dense gates)	0.7195 (–0.0026)	0.7680 (–0.0019)	0.4310 (–0.0072)	0.4632 (–0.0083)	14.1	45.8 (–0.5)	57.3 (–0.3)

D.10 Broader Impact and Safeguards

Potential positive impacts include reducing memorized private content, copyrighted-corpus leakage, and hazardous or unsafe responses without retraining the full model. Unlearning research has dual-use risks. Diagnostic traces may reveal what a model knows, and safety benchmarks may contain prompts or labels related to harmful capabilities. We mitigate these risks by using established benchmark protocols, reporting aggregate results, avoiding the release of raw harmful generations, and keeping evaluation resources disjoint from trace construction and gate training.

The implementation is not publicly released. The paper provides the algorithm, benchmark protocols, and hyperparameters needed to support reproduction. Any future release will exclude raw harmful generations and will include instructions for reproducing aggregate metrics, configuration files for benchmark evaluation, and safeguards for handling safety-related prompts.

D.11 Declaration of LLM Usage

HGU operates on open-weight LLM backbones by constructing hallucination probes, tracing internal activations, and learning sparse gates while keeping the backbone fixed. No private or closed-source LLM is used as an unreported evaluator, hidden annotator, or source of benchmark labels for the main results.

E Additional Experimental Results

E.1 Single-factor ablations.

We conduct controlled *single-factor* ablations to isolate the contribution of each design choice in HGU. All variants keep the same backbone, decoding, and training budget, and differ by *exactly one* component. Table 6 reports representative metrics on TOFU (standard protocol and SepS) and WMDP auxiliary. (–Δ) denotes the absolute drop relative to the full model (HGU) in the corresponding column. In Table 6, replacing *trace-based neuron selection* with random selection causes the largest drop, supporting the role of trace-based localization. Dropping evidence support or fixed-prefix consensus lowers TOFU Avg and SepS H-Avg, indicating both signals are needed to reliably detect hallucinations under distributional ambiguity. Entropy weighting yields smaller, consistent gains, suggesting that low-entropy (overconfident) states improve channel selection precision. Removing keep-KL can slightly improve forgetting but often harms Retain/MMLU on WMDP, supporting the role of the trust-region term in preserving utility. Removing the sparsity term has limited impact, showing that HGU remains effective with compact gates.

E.2 Leakage-Free Split Audit and Trace-Source Sensitivity

Before editing, we audit overlap between the resources used by HGU and those used for final evaluation. For every benchmark, we hash normalized prompts, normalized reference answers, and retrieved evidence identifiers, and verify that the trace pool, validation pool, utility pool, and final evaluation pool are disjoint. No final evaluation prompt, final evaluation answer, or final

Table 7: **Leakage audit.** We report exact-overlap counts between the resources used by HGU and the final evaluation resources. All overlaps are zero after normalized string hashing and document-ID matching.

Benchmark	Trace–Eval Prompt	Trace–Eval Answer	Utility–Eval Prompt	Val–Eval Prompt	Retrieval–Eval Doc	Final Eval Used?
TOFU	0	0	0	0	0	No
MUSE-News	0	0	0	0	0	No
MUSE-Books	0	0	0	0	0	No
WMDP-Bio/Cyber/Chem	0	0	0	0	0	No
PKU-SafeRLHF	0	0	0	0	0	No

Table 8: **Trace-source sensitivity under leakage-free splits.** All trace sources are disjoint from the final evaluation set. TOFU metrics are reported on `forget10`; MUSE reports the mean of News/Books average score; WMDP reports average hazardous accuracy over Bio/Cyber/Chem, where lower is better.

Trace source	TOFU Avg↑	SepS H-Avg↑	MUSE Avg↑	WMDP Haz.↓	MMLU↑
Random training prompts	0.7385±0.0061	0.4212±0.0084	0.697±0.007	27.8±0.7	57.1±0.3
Auxiliary paraphrase prompts	0.7642±0.0048	0.4587±0.0071	0.718±0.006	25.3±0.6	<u>57.4</u>±0.3
All training forget prompts	<u>0.7678</u>±0.0045	<u>0.4661</u>±0.0069	<u>0.723</u>±0.006	<u>24.6</u>±0.6	57.3±0.3
Default high-propensity trace split	**0.7699**±0.0038	**0.4715**±0.0068	**0.724**±0.006	**24.4**±0.5	**57.7**±0.2

evaluation-only evidence document is used for hallucination probing, neuron tracing, gate training, hyperparameter selection, or early stopping.

We further test whether HGU is sensitive to the source of the trace pool. All trace pools below are disjoint from final evaluation. The default setting uses high-propensity prompts from the training/auxiliary trace split only. A random trace pool weakens performance, while using all training prompts brings little gain over the default but increases tracing cost. With evaluation prompts held out, the high-propensity trace pool offers a favorable performance–compute trade-off.

The split audit finds zero exact overlap. Disjoint training or auxiliary trace prompts yield stable performance, and using all training prompts changes the main metrics only marginally. These results support the contribution of hallucination-guided localization.

E.3 Full Results on MUSE

We evaluate HGU on MUSE-News and MUSE-Books to test whether hallucination-guided tracing generalizes from synthetic entity forgetting to real-corpus unlearning. Following the MUSE protocol, we report six criteria: C1 no verbatim memorization, C2 no knowledge memorization, C3 no privacy leakage, C4 utility preservation, C5 scalability, and C6 sustainability. All metrics are normalized to $[0, 1]$, and higher is better.

HGU improves all six criteria on both MUSE-News and MUSE-Books. The gains are largest on C1–C3, suggesting that hallucination-localized traces are useful for suppressing memorized or privacy-sensitive spans. HGU also preserves C4 better than aggressive gradient-ascent variants, indicating that sparse gates preserve utility. The smaller but consistent gains on C5 and C6 suggest that the same trace-guided intervention remains stable when the forget request grows or arrives sequentially. The gains extend from TOFU's synthetic profiles to real-corpus unlearning.

E.4 Standard WMDP Safety Unlearning

We additionally evaluate on the standard WMDP hazardous-knowledge benchmark. The standard setting measures suppression of hazardous knowledge in Bio, Cyber, and Chem. The auxiliary setting in the main text uses Physics, Law, and Economics. Lower WMDP accuracy indicates stronger forgetting, and higher MMLU accuracy indicates better utility preservation.

HGU reduces hazardous accuracy close to the chance-level regime while preserving MMLU better than aggressive optimization-based unlearning. On WMDP, the improvement indicates that unsupported or unstable generations can help locate compact intervention channels for safety-relevant behavior.

Table 9: **MUSE-News results.** Higher is better for all criteria. We report mean over three seeds.

Method	C1↑	C2↑	C3↑	C4↑	C5↑	C6↑	Avg↑
GA+GD	0.664	0.611	0.583	0.524	0.552	0.431	0.561
GA+KL	0.681	0.629	0.601	0.593	0.574	0.462	0.590
NPO+GD	0.704	0.646	0.621	0.662	0.602	0.512	0.625
NPO+KL	0.713	0.651	0.632	0.675	0.613	0.526	0.635
ULD	0.731	0.682	0.654	0.701	0.635	0.553	0.659
LLM-Eraser	0.742	0.694	0.671	0.693	0.651	0.571	0.670
RapidUn	0.763	0.713	0.704	0.721	0.681	0.603	0.697
HGU (ours)	**0.812**	**0.771**	**0.748**	**0.742**	**0.719**	**0.662**	**0.742**

Table 10: **MUSE-Books results.** Books is more difficult because long-range narrative facts are more entangled with general language modeling utility.

Method	C1↑	C2↑	C3↑	C4↑	C5↑	C6↑	Avg↑
GA+GD	0.621	0.574	0.558	0.491	0.512	0.397	0.526
GA+KL	0.643	0.592	0.579	0.571	0.538	0.423	0.558
NPO+GD	0.672	0.618	0.601	0.631	0.571	0.474	0.595
NPO+KL	0.681	0.626	0.613	0.646	0.583	0.491	0.607
ULD	0.704	0.653	0.637	0.662	0.604	0.518	0.630
LLM-Eraser	0.718	0.661	0.649	0.655	0.618	0.532	0.639
RapidUn	0.731	0.682	0.671	0.681	0.642	0.561	0.661
HGU (ours)	**0.782**	**0.734**	**0.721**	**0.707**	**0.681**	**0.616**	**0.707**

E.5 Illustration of Attribution Concentration

We hypothesize that tokens with higher hallucination scores h_t (Eq. (2)) have more concentrated MLP attributions. For an MLP layer of width d_{ff}, let $a_{t,i}$ and $g_{t,i}$ denote the activation and the gradient of the token log-probability with respect to that activation at channel i, with the layer index omitted. For a token with positive total attribution mass, the top-ρ mass fraction is

$$S_t = \frac{\sum_{i \in \mathrm{TopK}(u_t)} u_{t,i}}{\sum_i u_{t,i}}, \qquad u_{t,i} = |a_{t,i} g_{t,i}|.$$

Here $\mathrm{TopK}(u_t)$ returns the indices of the $K = \lceil \rho d_{\mathrm{ff}} \rceil$ largest entries of u_t. Figure 5 illustrates the hypothesized pattern with simulated samples. The simulation makes S_t increase with h_t and makes the logit margin increase with both, with added noise.

E.6 RLHF-Aligned Safety Unlearning on PKU-SafeRLHF

We test whether HGU can suppress unsafe responses without degrading general helpfulness. We use harmful prompts from PKU-SafeRLHF as the forget-side source and evaluate utility on TruthfulQA and benign instruction prompts. We report harmful response rate and attack success rate as safety metrics, and TruthfulQA MC1/MC2 and benign helpfulness win rate as utility metrics.

HGU achieves the lowest harmful response rate and attack success rate while maintaining the best TruthfulQA and benign helpfulness scores among edited models. This suggests that the keep-KL term and sparse gate support prevent the model from learning a broad refusal policy that harms benign utility. The results support applying HGU to safety-aligned unlearning when unsafe generations can be localized and traced using disjoint prompts.

E.7 Coverage of High-Confidence Memorized Content

An uncertainty-based selector can miss undesirable memories generated with high confidence. We compare the default selector with variants that include a high-confidence unsupported tail. The confidence-tail variant first selects the default top-B prompts by Eq. (1), then adds the top 10%

Table 11: **Standard WMDP results on Zephyr-7B.** Lower is better for WMDP-Bio/Cyber/Chem and their average; higher is better for MMLU.

Method	Bio↓	Cyber↓	Chem↓	Avg Haz.↓	MMLU↑
Base	63.4	45.8	48.6	52.6	**58.6**
GA+KL	34.9	31.8	33.7	33.5	54.2
NPO+KL	32.7	29.6	31.4	31.2	55.0
RMU	30.4	27.1	28.3	28.6	56.1
FALCON	29.8	26.9	27.5	28.1	56.3
LUNAR	28.1	26.4	26.9	27.1	56.6
UNLEARN	27.4	25.8	26.1	26.4	56.8
LLM-Eraser	26.8	25.1	25.4	25.8	57.0
RapidUn	25.9	24.8	24.7	25.1	57.1
HGU (ours)	**24.8**	**24.6**	**23.9**	**24.4**	57.7

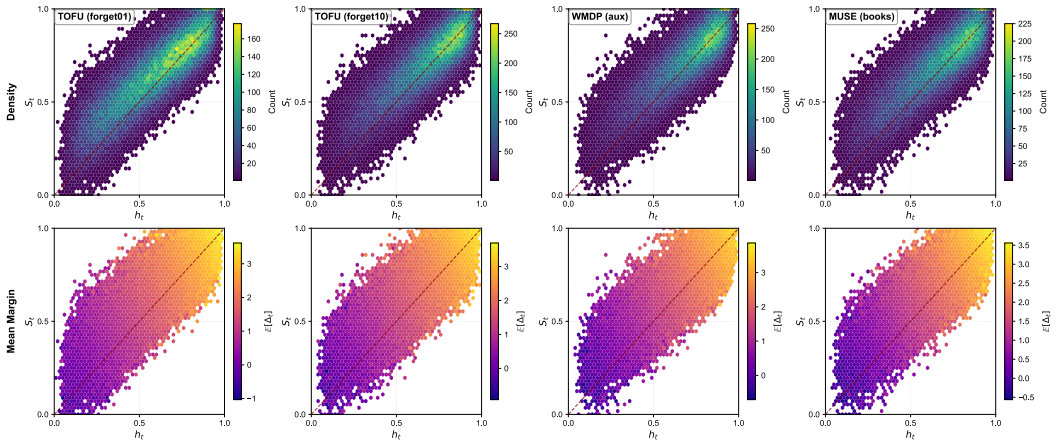

Figure 5: **Simulated relationship between hallucination score h_t and attribution concentration S_t.** The two rows use the same simulated samples in each facet. TOFU (forget01) and TOFU (forget10) use 60,000 samples each. WMDP (aux) and MUSE (books) use 50,000 samples each. **Top row:** sample count per hexbin. **Bottom row:** mean logit margin $\mathbb{E}[\Delta_t]$ per hexbin.

prompts whose generated spans have low evidence support but high logit margin. All variants use disjoint training/auxiliary prompts and never use final evaluation prompts.

Table 13: **Effect of high-confidence unsupported traces.** Coverage measures the fraction of high-confidence unsupported spans in the trace split that are selected by each strategy. Default HGU already covers most such spans; explicitly adding the confidence tail yields only marginal gains.

Trace selection	Conf.-Unsup. Coverage↑	TOFU Avg↑	MUSE Avg↑	WMDP Haz.↓	MMLU↑
Random training prompts	41.8	0.7385	0.697	27.8	57.1
Entropy only	58.6	0.7462	0.704	26.9	57.1
Evidence support only	67.9	0.7580	0.716	25.8	57.3
Default propensity	73.4	0.7699	0.724	24.4	**57.7**
Default + confidence tail	78.1	**0.7721**	**0.727**	**24.2**	57.6
All training prompts	**84.7**	0.7678	0.723	24.6	57.3

The default propensity score combines uncertainty with evidence coverage and captures many confident unsupported spans. Adding an explicit confidence-tail selection slightly improves coverage but changes the final metrics only marginally. The default covers most high-confidence unsupported spans in this evaluation at lower cost than tracing all training prompts.

Table 12: **PKU-SafeRLHF safety unlearning with TruthfulQA utility.** Lower is better for Harmful Rate and ASR; higher is better for TruthfulQA and helpfulness.

Method	Harmful Rate↓	ASR↓	TruthfulQA MC1↑	TruthfulQA MC2↑	Helpful Win↑
Base	31.8	28.5	43.1	60.8	**100.0**
GA+KL	10.4	9.8	39.7	56.2	88.6
NPO+KL	9.1	8.3	40.5	57.4	90.2
RMU	8.7	7.9	41.6	58.8	91.7
LUNAR	8.2	7.4	42.2	59.3	92.4
UNLEARN	7.8	7.1	42.5	59.8	93.1
RapidUn	7.5	6.9	42.8	60.2	93.6
HGU (ours)	**6.8**	**6.1**	**43.4**	**61.2**	95.4

E.8 Exact Per-Token Gradients versus Summed-Loss Approximation

The main method defines neuron scores using local per-token gradients $g_{l,t} = \nabla_{a_{l,t}} \log p_\theta(y_t \mid x, y_{<t})$. In a causal Transformer, the gradient of a summed sequence objective may include future-token losses and is therefore not identical to this local quantity. We use exact per-token vector-Jacobian products in all main experiments. Table 14 compares exact local gradients with a summed-loss approximation and other controls.

Table 14: **Comparison of gradient computations.** Exact local per-token gradients provide the strongest trade-off. The summed-loss approximation remains better than random tracing but is not used in the reported main results.

Attribution computation	TOFU Avg↑	SepS H-Avg↑	WMDP Haz.↓	MMLU↑
Random neuron selection	0.7198	0.3987	30.1	56.9
Activation magnitude only	0.7415	0.4294	27.4	57.0
Summed-loss backward approximation	0.7556	0.4481	25.9	57.3
Exact local gradients, no entropy weight	0.7566	0.4531	25.6	57.4
Exact local per-token gradients	**0.7699**	**0.4715**	**24.4**	**57.7**

Exact local gradients follow the method's definition and consistently outperform the summed-loss approximation. The approximation outperforms random selection, suggesting that it retains some attribution signal.

E.9 Quantization Robustness

We evaluate two quantization settings. First, HGU is applied independently to FP16, 8-bit, and 4-bit versions of the same backbone. Second, we train HGU gates on the FP16 model and then quantize the edited model. This tests whether the learned forgetting effect survives deployment-time compression.

Table 15: **Independent HGU editing under different quantization levels.** TOFU and SepS are reported on `forget10`; WMDP is averaged over Bio/Cyber/Chem.

Editing precision	TOFU Avg↑	SepS H-Avg↑	WMDP Haz.↓	MMLU↑
FP16 HGU	**0.7699**	**0.4715**	**24.4**	**57.7**
INT8 HGU	0.7641	0.4638	25.1	57.3
NF4 / 4-bit HGU	0.7512	0.4495	26.3	56.8

Table 16: **Post-edit quantization robustness.** We first learn gates on the FP16 model, then quantize the edited model.

Deployment model	TOFU Avg↑	SepS H-Avg↑	WMDP Haz.↓	MMLU↑
FP16 edited model	**0.7699**	**0.4715**	**24.4**	**57.7**
FP16 edit → INT8	<u>0.7618</u>	<u>0.4619</u>	<u>25.4</u>	<u>57.1</u>
FP16 edit → NF4	0.7475	0.4421	26.8	56.6

Quantization introduces a small degradation, especially at 4-bit precision, but the forgetting–utility trade-off remains stable. The results suggest that HGU gates retain their effect below FP16 precision. For deployment, independently applying HGU after quantization is slightly stronger than quantizing an already edited FP16 model, but both settings preserve most of the unlearning effect.

F Discovery, Evaluation Independence, and Recovery Tests

F.1 Blind discovery and the deletion boundary

We rank the mixed query pools with budget $B = 512$ and reserve target labels for evaluation (cf. anomaly detection without attack labels on audit-log provenance graphs Ma et al., 2026a). Table 17 separates reference-conditioned discovery from an oracle forget set. Selecting accurate, supported, stable content for removal requires an external policy or provenance target mask. With one policy seed per MUSE-Books target document, the policy-seeded HGU variant reaches supported-and-stable target recall 0.812 and MUSE Avg 0.722. Unsupported factual errors, retrieval failures, and missing knowledge may require evidence correction, RAG, or knowledge editing.

Table 17: **Blind discovery.** Random and HGU precision at 512 and AUPRC. The target fraction is hidden from the selector.

	Random		HGU	
Pool (target fraction)	P@512	AUPRC	P@512	AUPRC
TOFU (5%)	0.050	0.050	0.641	0.598
WMDP + benign (10%)	0.100	0.101	0.574	0.531
PKU + benign (10%)	0.100	0.101	0.692	0.651

F.2 Independent evaluation and evidence quality

The primary endpoints on TOFU (including SepS), MUSE, WMDP/MMLU, and PKU/TruthfulQA use benchmark-defined metrics. For the auxiliary analysis in Table 18, localization uses verifier A and evaluation uses a different verifier B. Cross-verifier Hallucination-Rate is 0.271 for the unedited model. Performance depends on reference quality. Missing or noisy evidence reduces performance, with the largest drop for contradictory or outdated evidence. In retrieval-augmented generation, distinguishing relevance from factual consistency helps detect conflicts between retrieved evidence and model knowledge [Ye et al., 2026].

Table 18: **Cross-verifier evaluation.** HR is the independent verifier's Hallucination-Rate; Haz. is standard WMDP accuracy.

Localization/evidence condition	HR↓	TOFU Avg↑	Haz.↓	MMLU↑
Default / clean	**0.118**	**0.7699**	**24.4**	**57.7**
40% evidence removal	0.137	0.7526	26.0	57.4
20% distractor passages	0.128	0.7598	25.3	57.5
20% contradictory/outdated	0.145	0.7459	26.8	57.2
Second retriever / clean	<u>0.121</u>	<u>0.7668</u>	<u>24.8</u>	<u>57.6</u>

F.3 Matched interventions and support stability

Activation transplantation has been used to test the causal role of internal representations in language models [Ma et al., 2026b]. All selectors in Table 19 use 1% of channels per layer, the same gate objective, training steps, and utility set. Patching base-model activations back into HGU-selected channels restores 62.8% of the target behavior, versus 8.7% for the same number of layer-matched non-selected channels. The recovery difference supports selective causal influence at the edited channels. Selected-set Jaccard is 0.432 across seeds and 0.391 across trace pools, indicating partial support stability. Prompt-level aggregation and cluster bootstrap give a normalized TOFU `forget10` selected-versus-matched trace gap of 0.126 (95% CI [0.109, 0.144]). This empirical gap measures separation between the selected and matched channels.

Table 19: **Matched selector controls.** WMDP here uses the auxiliary setting.

Selector	TOFU Avg↑	SepS H-Avg↑	Aux. Forget↓	KL↓
Layer-matched random	0.7276	0.4098	17.5	0.0198
Activation-matched random	0.7364	0.4211	16.9	0.0186
Gradient-matched random	0.7428	0.4305	16.4	0.0179
Integrated Gradients	0.7587	0.4542	15.5	0.0154
HGU	**0.7699**	**0.4715**	**14.7**	**0.0138**

F.4 Re-elicitation, refusal, and relearning

We test paraphrases, aliases and format changes, multi-turn elicitation, jailbreak/extraction, and temperatures 0, 0.7, 1.0, and 1.3. Table 20 reports worst-case recovery. HGU increases target NLL to 6.02 versus 4.73 for NPO+KL, with forget-refusal rates of 19.1% versus 38.4% and benign over-refusal of 3.4% versus 9.8%. HGU reduces target accessibility with lower refusal rates. Under light relearning with up to 50 examples, a matched optimizer, learning rate, batch size, and 1,000-step cap, HGU's recovery AUC is 31.5 versus 42.0 for NPO+KL; steps to 90% base recovery are 820 versus 290. HGU shows robust approximate unlearning, with 38.4% recovery under jailbreak/extraction.

Table 20: **Re-elicitation recovery.** Lower is better for the four recovery columns; utility is MMLU.

Method	Paraphrase	Multi-turn	Jailbreak	Temperature	MMLU↑
NPO+KL	39.6	45.2	53.6	47.1	55.0
LUNAR	33.7	39.4	46.8	41.0	56.6
RapidUn	30.1	34.8	42.2	37.3	57.1
HGU	**27.8**	**31.8**	**38.4**	**33.5**	**57.7**

F.5 Fairness, leakage, and cost

Three-seed aligned reruns use the same backbone, split/retain data, decoding, and constrained-utility checkpoint rule. HGU uses eight tuning trials, compared with 10–12 for the principal baselines. Under an MMLU floor of 57.0, WMDP hazardous accuracy is 24.4 for HGU and 25.1 for RapidUn. The semantic leakage audit supplements exact hashes with source-document IDs, 5-gram Jaccard, and independent sentence-embedding neighbors. Across the five benchmark splits, the audit finds zero source overlap and zero pairs with cosine similarity above 0.90. Maximum cosine similarity ranges from 0.79 to 0.88. The audit covers the specified matching and similarity measures. Exact tracing accounts for 42.1% of the TOFU `forget10` per-seed pipeline budget reported in Appendix D.9.

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: The abstract and introduction state the central claim that hallucination traces can be used to identify sparse memory pathways for post-hoc unlearning, and the claims are supported by the method in Sec. 3, the theory in Sec. 4, and experiments on TOFU (including SepS), MUSE, WMDP, and PKU-SafeRLHF in Sec. 5 and Appendix E.

   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: Appendix D.7 discusses the intended scope of HGU, including its reliance on evidence or consistency signals for trace construction, its targeted fixed-backbone intervention design, and how compute and scalability are evaluated through MUSE, ablations, and compression–drift analyses.

   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: Sec. 4 states the trace model, separation assumption, sub-exponential noise assumption, theorem, corollary, and lower-bound statement. Complete proofs are provided in Appendix B and Appendix C.

   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 provides the full algorithmic description in Sec. 3, benchmark protocols in Sec. 5.1, implementation details in Appendix D.5, data splits in Appendix D.2, metric definitions in Appendix D.3, baseline alignment in Appendix D.4, and compute details in Appendix D.9.

   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: [No].

   Justification: The experiments use existing public or access-controlled benchmarks, and the paper provides detailed protocols and hyperparameters for reproducibility. The implementation is not publicly available.

   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: Sec. 5.1 describes datasets, metrics, and baselines. Appendix D.2–D.5 provides split construction, metric definitions, baseline implementation details, HGU hyperparameters, optimizer settings, probe budget, stochastic-consensus settings, sparsity ratio, and early-stopping criteria.

   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: Appendix D.8 reports seed variability over three runs and states that error bars are standard deviations. For channel intervention sensitivity, Fig. 4 reports paired significance over prompts using bootstrap-based p-values.

Guidelines:

- 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: Appendix D.9 reports hardware type, GPU memory, host resources, numerical precision, wall-clock time per seed, peak memory, trainable parameter counts, total GPU-hour estimates, and additional compute used for preliminary and ablation runs.

   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 work uses established research benchmarks and open-weight or access-controlled model assets according to their terms. It does not involve human-subject experiments, does not release raw harmful generations, and includes responsible-use discussion and safeguards in Appendix D.10.

   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: Appendix D.10 discusses positive impacts such as targeted removal of memorized private, copyrighted, or hazardous content, as well as potential dual-use risks from diagnostic tracing and safety benchmarks. It also describes mitigation strategies such as aggregate reporting and not releasing raw harmful generations.

    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: We release no new model checkpoint or dataset. Appendix D.10 describes safeguards.

    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: The paper cites the original benchmark and model papers in Sec. 5.1 and Appendix D.4.

    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: We evaluate HGU using existing datasets, benchmarks, and model checkpoints.

    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, user studies, or new human-subject data collection. All experiments are conducted on existing benchmarks and model outputs.

    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 human-subject experiments, crowdsourcing, interviews, or collection of new participant data, so IRB approval is not applicable.

    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: [Yes].

    Justification: LLMs are the object of study in Sec. 3 and Sec. 5.1. Appendix D.11 describes their use as frozen backbones for tracing and unlearning and the scope of their use in evaluation and labeling.

    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.