Graph-Guided Selective Unlearning for Language Models: Controlling Support Routes Beyond Forget Seeds

Waqas Khan[1,✉], Tabinda Sarwar[1], Jingyue Cong[1], Xun Yi[1], and Estrid He[1,✉]
[1]RMIT University, Melbourne, Australia

✉ **Corresponding author contact:**, mwaqaskh@gmail.com
estrid.he@rmit.edu.au

Abstract

Enterprises fine-tune language models on proprietary data that may later require removal due to privacy, contractual, or compliance obligations. Selective unlearning removes requested knowledge while preserving model utility, offering a practical alternative to full retraining, but existing methods treat the explicitly identified forget examples as the complete deletion scope. This is insufficient when target knowledge remains recoverable through paraphrases, aliases, or neighboring training examples. We propose GRAPHSU, a graph-guided controller that expands the deletion scope beyond forget seeds by constructing a weighted support-route graph, propagating deletion pressure through it, and applying graded forgetting strengths to high-risk neighbors. On the Task of Fictitious Unlearning (TOFU), a synthetic author-profile question–answering benchmark, and PISTOL, a structural-unlearning benchmark built around interconnected factual samples, with GPT-2 Medium and Llama-3.2-3B-Instruct, GRAPHSU achieves the lowest utility-feasible soft leakage across all deletion settings, reducing leakage by up to 49.5 percentage points over a matched seed-only baseline, demonstrating that effective enterprise unlearning requires controlling support routes, not just forget seeds.Our code is available in our GitHub repository.

1 Introduction

Large language models (LLMs) can retain private, copyrighted, stale, or user-rescinded information after training. Exact unlearning would retrain the model on the corpus with the deletion set removed, but this is typically infeasible for modern LMs. Approximate LLM unlearning therefore aims to update a deployed model so that target knowledge is no longer recoverable, while its behaviour on retained data remains close to the original or retain-only model (Maini et al., 2024; Shi et al., 2024; Tian et al., 2024; Li et al., 2024).

As enterprises increasingly customize foundation models using proprietary organizational data, the ability to selectively remove previously learned information becomes critical. Data may later require removal due to privacy regulations, customer offboarding, intellectual-property concerns, policy revisions, or the discovery of sensitive content in training corpora. In such settings, machine unlearning provides a scalable mechanism for updating deployed models without incurring the computational and operational costs of full retraining.

Most current algorithms focus on the *local objective*: gradient ascent, KL-regularised ascent, preference optimisation, negative preference optimisation, gradient rectification, energy-based unlearning, or representation control (Jang et al., 2023; Zhang et al., 2024; Mekala et al., 2025; Wang et al., 2025b; Chaiyapattanaporn et al., 2026). This line of work is valuable, but it leaves a complementary question under-addressed: *where should unlearning be applied?* In selective unlearning, targeting only the canonical forget example is often insufficient, as the forgotten information may still be inferred from aliases, paraphrases, neighboring facts, or other training instances that describe the same entity from different perspectives. For example, forgetting *"Acme Corp plans to acquire BetaAI for $850 million"* might be insufficient if another sample states, *"Acme's largest strategic investment this year is valued at approximately $850 million"*. Conversely, applying unlearning too broadly can remove information that should be retained, leading to unnecessary degradation of model utility. This creates a scope-control problem that is distinct from the choice of unlearning objective.

In this paper, we formulate selective unlearning as a leakage-route control problem, where the goal is to prevent forgotten information from being reconstructed through paraphrases, aliases, and semantically related training instances while preserving unrelated knowledge. We propose **Graph-**

EMNLP 2026 Industry Track, pages 4398–4414
October 24-29, 2026 ©2026 Association for Computational Linguistics

guided **S**elective **U**nlearning (GRAPHSU), a loss-agnostic controller that determines the fine-grained deletion scope before applying a local unlearning objective. GRAPHSU constructs a weighted support graph over training instances using semantic similarity, symbolic entity/relation/tail overlap, and answer-side gradient alignment. It then propagates deletion pressure from forget seeds, identifies influential support neighbors, assigns soft forgetting strengths, and incorporates graph-selected neighbors through a staged unlearning curriculum. On GPT-2 Medium, GRAPHSU achieves the lowest utility-feasible soft leakage across all deletion settings on both TOFU and PISTOL. Compared to a Seed-Only variant, it consistently reduces leakage while maintaining retain perplexity (PPL; the exponential of mean token-level negative log-likelihood, lower is better) below 10, demonstrating the importance of expanding beyond the initial forget set.

2 Methodology

Figure 1 illustrates proposed method.

2.1 Problem Statement

Let f_{θ_0} denote the original model trained on dataset $D = \{z_i\}_{i=1}^{N}$ where $z_i = (q_i, a_i^{\text{text}}, \mathcal{F}_i)$. Here, q_i is the prompt, a_i^{text} is the answer surface form, and $\mathcal{F}_i = \{(h_{i,k}, r_{i,k}, t_{i,k})\}_{k=1}^{K_i}$ is the set of factual units expressed in the answer. Each tuple $(h_{i,k}, r_{i,k}, t_{i,k})$ consists of head entity, relation, and tail entity or value. We obtain $\mathcal{F}_i$ by prompting GPT-4.1 (OpenAI, 2025) to extract factual predicate-argument tuples from the prompt–answer pair and normalizing the output into (h, r, t) unit. The extracted units are used for target specification and support-graph construction, not as additional supervision for model training.

An unlearning request $\mathcal{R}$ identifies target fact units $\mathcal{K}_{\mathcal{R}}$ to remove and a seed forget set $D_f \subset D$ containing them; the initial retain pool is $D_r = D \setminus D_f$. The goal of selective unlearning is to obtain an updated model f_{θ_u} in which the knowledge specified by $\mathcal{K}_{\mathcal{R}}$ is no longer recoverable, while the model's utility on unrelated retain data is preserved.

2.2 Recovery-Aware Deletion Scope

As discussed in Section 1, the seed forget set D_f may miss related training instances that still support recovery of the target knowledge, leaving leakage routes outside D_f. We denote by $\mathcal{M}_{\mathcal{R}}(f_\theta)$ a scalar recoverability measure for $\mathcal{K}_{\mathcal{R}}$ under request-specific probes, capturing the extent to which the model can still reproduce, paraphrase, or semantically entail the target facts.

For any training example z_j, define its ideal support contribution as

$$\Delta_j^\star(\mathcal{R}) = \mathcal{M}_{\mathcal{R}}(f_{\theta_0}) - \mathcal{M}_{\mathcal{R}}(f_{\theta_0}^{(-j)}), \quad (1)$$

where $f_{\theta_0}^{(-j)}$ denotes the model obtained by applying the same supervised fine-tuning procedure after removing z_j from the training set.

We define the ideal support closure of $\mathcal{R}$ as

$$\mathcal{C}_\epsilon^\star(\mathcal{R}) = D_f \cup \{z_j \notin D_f : \Delta_j^\star(\mathcal{R}) \geq \epsilon\}, \quad (2)$$

where $\epsilon > 0$ is a small operational threshold. This closure is the ideal forgetting scope: it includes both the seed forget examples and the non-seed examples whose removal would materially reduce recovery of $\mathcal{K}_{\mathcal{R}}$. Constructing $\mathcal{C}_\epsilon^\star(\mathcal{R})$ exactly is infeasible, because it would require leave-one-out retraining or exact causal attribution over the training corpus. GRAPHSU therefore approximates this closure using a **request-conditioned support graph**, introduced next.

2.3 Support Graph Construction.

The seed forget set D_f is not generally a closed evidence unit. Even after examples in D_f are suppressed, the target knowledge may remain recoverable through **paraphrastic** (Maini et al., 2024; Reimers and Gurevych, 2019; Reisizadeh et al., 2025), **symbolic** (Tian et al., 2024; Wang et al., 2025a; Wei et al., 2025; Meng et al., 2022; Zhong et al., 2023; Cohen et al., 2024), and **model-internal** support routes (Koh and Liang, 2017; Pruthi et al., 2020; Park et al., 2023). We therefore construct a weighted support graph $G = (V, E, W)$, where each node corresponds to a training example and each edge weight estimates how strongly one example may support recovery of another. The graph is used as an affinity structure for support expansion, not as a calibrated estimate of causal influence.

Multi-view support affinity. For a candidate edge (i, j), GRAPHSU fuses five support views three symbolic (entity, relation, tail), one semantic, and one gradient-based:

$$\begin{aligned} \tilde{w}_{ij} &= \alpha s_{ij}^{\text{ent}} + \beta s_{ij}^{\text{rel}} + \gamma s_{ij}^{\text{tail}} + \delta s_{ij}^{\text{sem}} + \lambda s_{ij}^{\text{grad}}, \\ w_{ij} &= \min\big(1, \max(0, \tilde{w}_{ij})\big), \end{aligned} \quad (3)$$

where each component score is normalized to $[0, 1]$ and $(\alpha, \beta, \gamma, \delta, \lambda)$ are non-negative view weights

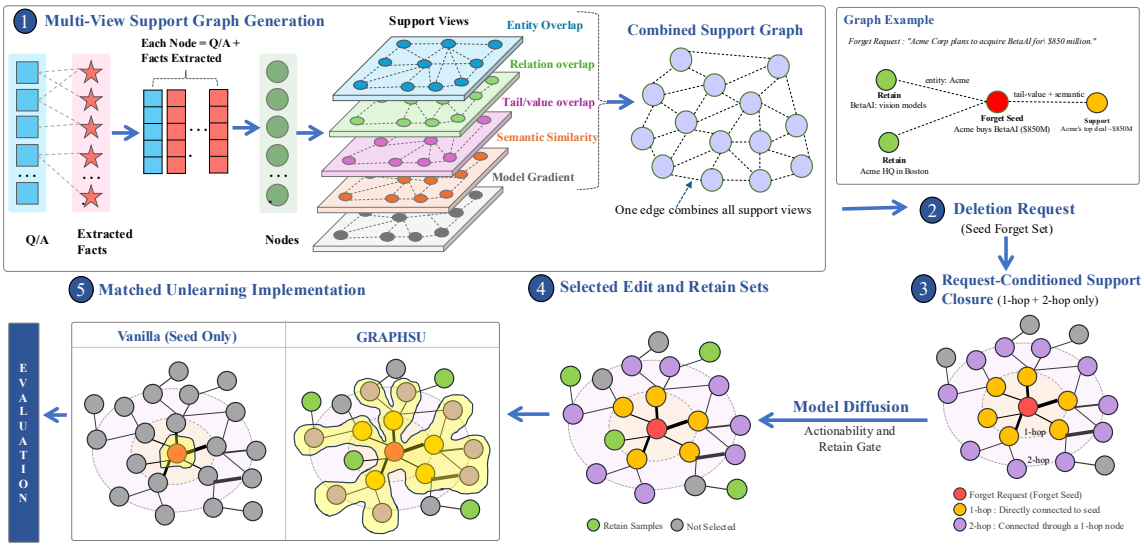

Figure 1: Overview of GRAPHSU, which constructs a weighted multi-view support graph over QA samples, propagates deletion pressure from forget seeds through a one- and two-hop support closure, and applies graded unlearning objectives to the expanded edit set.

(Table 1). The five views capture complementary recovery routes: the three symbolic overlaps capture structured factual support through shared entities, relations, and tail values; semantic similarity captures paraphrastic support; and gradient alignment captures model-local support under the frozen post-SFT model.

Semantic evidence. Let $e_i = \mathrm{Enc}([q_i; a_i])$ be the ℓ_2-normalized sentence embedding of the prompt–answer pair. The semantic score is

$$s_{ij}^{\mathrm{sem}} = [\cos(e_i, e_j)]_+ , \qquad [x]_+ = \max(0, x). \tag{4}$$

Negative semantic cosine indicates dissimilar rather than supporting content. Because GRAPHSU models positive recovery support with a non-negative diffusion process, negative semantic affinities are clipped to zero rather than allowed to subtract from supportive edges. Modelling inhibitory relationships would require a signed-graph formulation and is left for future work.

Symbolic evidence. Semantic similarity can miss structured dependencies such as aliases, inverse relations, and reused private values. We extract canonicalized entity, relation, and tail-value sets $\mathcal{E}_i$, $\mathcal{R}_i$, and $\mathcal{T}_i$ from q_i, a_i, and $\mathcal{F}_i$ respectively. Entity aliasing and relation canonicalization map equivalent mentions and inverse relations to shared identifiers. We then compute entity-, relation-, and tail-level symbolic similarities s_{ij} between instances i and j using Jaccard similarity over the corresponding sets.

Entity overlap (s^{ent}) captures subject- and alias-level support, relation overlap (s^{rel}) captures factual-route support, and tail-value overlap (s^{tail}) captures reused attributes or identifiers.

Gradient-alignment evidence. The semantic and symbolic views are corpus-level approximations. To capture model-local support, we compute answer-side gradient summaries under the frozen post-SFT model. Let $\mathcal{A}_i$ denote the answer-token positions of z_i. We define

$$g_i = \frac{1}{|\mathcal{A}_i|} \sum_{t \in \mathcal{A}_i} \nabla_{H_{it}^{(L)}} \mathcal{L}_{\mathrm{CE}}(z_i; \theta_0), \tag{5}$$

where $H_{it}^{(L)}$ is the final-layer answer representation after LayerNorm and before the LM head. The gradient-alignment score is

$$s_{ij}^{\mathrm{grad}} = [\cos(g_i, g_j)]_+ . \tag{6}$$

Positive gradient cosine indicates compatible, mutually reinforcing local update directions, whereas negative cosine indicates conflicting directions rather than recovery support. Since GRAPHSU propagates non-negative support mass, negative gradient affinities are clipped to zero; representing inhibitory relations would instead require signed-graph propagation.

The graph captures recovery-aware support structure around the request. GRAPHSU propagates deletion pressure from D_f through G, assigning graded forgetting strengths to high-risk neighbors while leaving unrelated examples intact.

2.4 Graph-Guided Deletion Scope Approximation

Given an unlearning request $\mathcal{R}$ with the seed forget set D_f. We first estimate which non-seed examples are most likely to belong to the recovery-aware deletion scope of the request.

Let A denote the non-negative adjacency matrix of the sparsified support graph. We use the pairwise affinity w_{ij} from Eq. 3 directly, so

$$A_{ij} = \begin{cases} w_{ij}, & (i,j) \in E \text{ after sparsification,} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Because the component similarities in Eq. 3 are pairwise symmetric, $w_{ij} = w_{ji}$ before sparsification; the retained graph is treated as undirected for diffusion. Let $d_i = \sum_j A_{ij}$ be the weighted degree of node i. We define the row-normalized transition matrix

$$P_{ij} = \begin{cases} A_{ij}/d_i, & d_i > 0, \\ \mathbb{I}[i=j], & d_i = 0, \end{cases} \tag{8}$$

where isolated nodes are assigned self-loops. The transition matrix specifies how support evidence moves across the graph: a node passes more evidence to neighbors connected by stronger semantic, symbolic, or gradient-aligned support edges. We initialize a request-specific seed distribution

$$s_i = \begin{cases} 1/|D_f|, & z_i \in D_f, \\ 0, & z_i \notin D_f. \end{cases} \tag{9}$$

The support score vector r is then computed by personalized graph diffusion:

$$r^{(k+1)} = (1-\rho)P^\top r^{(k)} + \rho s, \tag{10}$$

where $\rho \in (0,1)$ is the restart probability. The restart term anchors the diffusion to the deletion request, while the transition term propagates support evidence through the graph. At convergence, the solution satisfies

$$r = \rho \left(I - (1-\rho)P^\top\right)^{-1} s, \tag{11}$$

or equivalently,

$$r = \rho \sum_{\ell=0}^{\infty} (1-\rho)^\ell (P^\top)^\ell s. \tag{12}$$

This expansion shows that r_i aggregates support evidence over all paths from the seed forget examples to z_i, with paths of length ℓ discounted by $(1-\rho)^\ell$. Hence, r_i is high when z_i is connected to D_f through many strong short paths, or through multiple consistent longer paths. The score is not interpreted as a probability of membership in the deletion scope; it is a ranking signal for request-specific support evidence.

The estimated support closure is obtained by selecting high-scoring non-seed examples together with the original seeds. For a threshold η or an expansion budget B, we use

$$\widehat{\mathcal{C}}_\eta(\mathcal{R}) = D_f \cup \{z_i \in D_r : r_i \geq \eta\}, \tag{13}$$

or equivalently the top-B non-seed examples ranked by r_i when a fixed expansion budget is required. This construction approximates the ideal support closure $\mathcal{C}^\star_\epsilon(\mathcal{R})$ defined in Section 2.2: examples with larger diffusion scores are treated as more likely to materially support recovery of the target knowledge.

2.5 Selective Unlearning

Let $D_{sup}(\mathcal{R})$ be the selected non-seed support set:

$$D_{\text{sup}}(\mathcal{R}) = \widehat{\mathcal{C}}_\eta(\mathcal{R}) \setminus D_f. \tag{14}$$

Each example receives a forgetting weight

$$\omega_i = \begin{cases} 1, & z_i \in D_f, \\ \eta_{\mathcal{R}} \left(\dfrac{r_i}{\max_{z_j \in D_{\text{sup}}} r_j} \right)^{\gamma_\omega}, & z_i \in D_{\text{sup}}(\mathcal{R}), \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Thus, seed examples receive full forgetting pressure, while non-seed support examples receive bounded pressure according to graph relevance. Let m_i be a binary answer-token mask, with $m_{i,t} = 1$ for target tokens to be forgotten and $m_{i,t} = 0$ for answer tokens that should be retained. The complement $1 - m_i$ therefore selects the safe answer tokens. The unlearning objective is

$$\begin{aligned} \mathcal{L}_{\text{GraphSU}} = \sum_{i=1}^{N} \Big[& \omega_i \mathcal{L}_{\text{forget}}(z_i, m_i) \\ & + \lambda_r \mathcal{L}_{\text{retain}}(z_i, 1-m_i) \Big] \\ & + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}. \end{aligned} \tag{16}$$

The forgetting term is applied only to the target positions selected by m_i and combines three components:

$$\mathcal{L}_{\text{forget}} = \lambda_{\text{ent}} \mathcal{L}_{\text{ent}} + \lambda_{\text{UL}} \mathcal{L}_{\text{UL}} + \lambda_{\text{rep}} \mathcal{L}_{\text{rep}}. \tag{17}$$

Here, $\mathcal{L}_{\text{ent}}$ is the entropy-maximization term that flattens the predictive distribution at target tokens, $\mathcal{L}_{\text{UL}}$ is the unlikelihood term that suppresses probability assigned to the target tokens, and $\mathcal{L}_{\text{rep}}$ is

the representation-repulsion term used to move target-token representations away from their frozen post-SFT representations. The coefficients λ_{ent}, λ_{UL}, and λ_{rep} control the relative strengths of these terms. The retain loss $\mathcal{L}_{\text{retain}}$ is cross-entropy on the complementary safe-token mask $1 - m_i$, while $\mathcal{L}_{\text{KL}}$ anchors non-target predictive behaviour to the frozen post-SFT model; λ_r and λ_{KL} are their corresponding weights. Representation repulsion is enabled for complete and entity/instance deletion and disabled for partial deletion, where target and non-target facts may share the same answer context. In this way, GRAPHSU separates the scope controller from the local unlearning loss: the graph determines which examples and tokens are edited, while the loss defines how the selected targets are suppressed.

3 Experiments

3.1 Experimental Settings

Datasets and Models. We evaluate GRAPHSU on two unlearning benchmarks: TOFU (Maini et al., 2024), a synthetic author-profile QA benchmark with auditable forget/retain splits, and PISTOL (Qiu et al., 2024), a structural-unlearning benchmark built around interconnected factual samples. We consider complete, entity/instance, and partial deletion settings. We use GPT-2 Medium for the main controlled comparison and LLAMA-3.2-3B-INSTRUCT (`meta-llama/Llama-3.2-3B-Instruct`) to assess scalability to larger backbones. For each model, all methods are initialized from the same supervised fine-tuned checkpoint and evaluated using identical forget seeds and prompts.

Baselines. We compare GRAPHSU with existing methods including GA (Jang et al., 2023), GA+KL (Maini et al., 2024), DPO (Rafailov et al., 2023), NPO-TR (Zhang et al., 2024), AltPO (Mekala et al., 2025), GRU (Wang et al., 2025b), and SELU (Chaiyapattanaporn et al., 2026). NPO-TR denotes an NPO-style baseline with an explicit retain-term regularizer. We also compare GRAPHSU with Seed-Only, a controlled ablation that shares the same unlearning objective as GRAPHSU but removes graph-guided controller. Key implementation details are summarized in Table 1.

Table 1: Core implementation settings. Full hyperparameter sweeps are deferred to supplementary material.

Setting	Complete	Entity	Partial
Steps	1200	1200	720
Graph activation s_g	120	120	220
Neighbour cap K_n	72	72	40
Neighbour $\phi^{(n)}_{max}$	0.35	0.32	0.15
Entropy weight λ_{ent}	1.35	1.40	1.15
Unlikelihood weight λ_{UL}	0.60	0.60	0.25
Retain CE weight λ_r	0.22	0.20	0.24
KL weight λ_{KL}	0.01	0.01	0.012
Repulsion enabled	on	on	off
Route prompts per seed	4	5+aliases	6
Edge weights $(\alpha, \beta, \gamma, \delta, \lambda)$	$(0.7, 0.5, 0.5, 1.0, 0.7)$		

3.2 Metrics and Route Probes

The primary forget metric is soft leakage. For each forget seed $z_i \in D_f$, let $\{p_{i,k}\}_{k=1}^{K_i}$ denote its validated recovery-route prompts, where k indexes an applicable route prompt and K_i is the number of valid prompts for that seed after route generation and filtering (Appendix E). Let $y'_{i,k}$ be the model continuation generated from $p_{i,k}$, and let target_i denote the sensitive answer string or grounded target value whose recoverability is being tested for seed i. We define $s_{i,k} = \text{sim}(y'_{i,k}, \text{target}_i)$ as the sentence-embedding cosine similarity between the generated continuation and that target. Soft leakage is

$$\text{SOFTLEAK} = \frac{1}{|D_f|} \sum_{z_i \in D_f} \frac{1}{K_i} \sum_{k=1}^{K_i} \mathbb{1}[\, s_{i,k} \geq \tau_{\text{soft}} \,], \tag{18}$$

where $\tau_{\text{soft}} = 0.85$ is the similarity threshold used to count a route as leaking the target. For both datasets, we evaluate three deletion settings: complete deletion, where the target is the full answer; partial deletion, where the target is a set of sensitive spans; and entity deletion, where the target is the canonical entity profile together with its aliases. Retention is measured using retain perplexity (PPL), retain F1, ROUGE-L F1, and separate retain-neighbor, retain-far, and partial-retain-only panels. PPL is the exponential of the mean token-level negative log-likelihood, so lower values indicate better retained language-modelling performance. Appendix I reports the additional utility panels jointly with soft leakage. Route families include direct, paraphrase, indirect, cloze, related-fact, list-sum., relation-inversion, and entity-alias prompts.

3.3 Main Results

Utility-constrained unlearning. Table 2 reports the main GPT-2 Medium results on TOFU and PISTOL; results with LLAMA-3.2-3B-INSTRUCT are provided in Appendix A. Across both datasets and all deletion settings, GRAPHSU achieves the lowest soft leakage among methods that keep retain PPL below 10. This distinction is important because effective unlearning requires balancing forgetting and retention. DPO, AltPO, and SELU preserve low retain PPL but leave high leakage, while GA, GA+KL, and GRU can reduce leakage only with substantial retention loss. For example, on TOFU complete deletion, GRU obtains lower leakage than GRAPHSU (38.13% vs. 46.83%) but raises retain PPL to 79.11, making it unsuitable under the utility-feasible criterion.

Impact of graph-guided support expansion. Since Seed-Only uses the same unlearning objective but omits graph-neighbor expansion, the comparison isolates the effect of support-route control in GRAPHSU. On TOFU, GRAPHSU reduces aggregate soft leakage by 46.42 points for complete deletion, 41.65 points for entity/instance deletion, and 18.33 points for partial deletion. On PISTOL, the corresponding reductions are 49.50, 21.66, and 12.22 points. Figure 2 further shows that the improvement is consistent across recovery-route families: GRAPHSU improves over Seed-Only in all 21 task–route cells, with average route-level reductions of 45.9, 41.7, and 22.6 points for complete, entity/instance, and partial deletion, respectively. These results show that effective unlearning requires deleting support routes, not just seed examples. Appendix I complements PPL with retain F1, ROUGE-L F1, far-retain F1, neighbour-retain F1, and safe-span F1 for partial deletion. On TOFU, GRAPHSU raises retain F1 from 39.07 to 42.41 for complete deletion, from 37.16 to 59.11 for entity/instance deletion, and from 3.20 to 55.14 for partial deletion; under partial deletion, safe-span F1 rises from 1.26 to 61.31. These answer-level results show that the controlled Seed-Only gains reflect an improved forgetting–utility balance rather than leakage reduction in isolation.

Challenges of partial deletion. Complete and entity/instance deletion benefit most from graph expansion because target information can be supported through full answers, aliases, and related facts. Partial deletion is more constrained: the model must suppress sensitive spans while preserving the remaining answer content. Although GRAPHSU improves the partial-deletion frontier, achieving 81.67% leakage with retain PPL 1.91 on TOFU, substantial residual leakage remains. These results support graph-guided support expansion and route-level control, while stronger claims about certified erasure require further ablation.

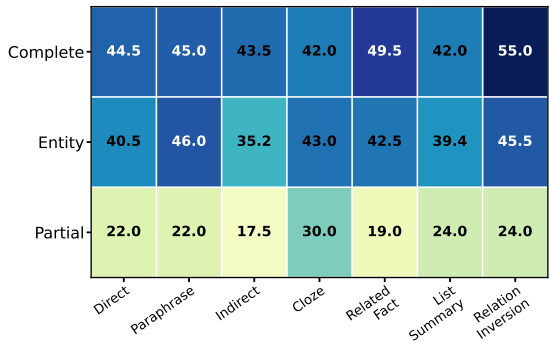

Figure 2: Route-level soft-leakage reduction of GRAPHSU relative to Seed-Only on GPT-2 Medium (TOFU). Values denote percentage-point reductions in soft leakage; higher is better.

3.4 Recovery-route robustness

Figure 2 reports the route-level soft-leakage reduction of GRAPHSU relative to Seed-Only across seven recovery-route families and three deletion settings. GRAPHSU improves over Seed-Only in all 21 task–route combinations, reducing leakage by an average of 36.8 percentage points. Gains are largest for complete deletion, with an average reduction of 45.9 points and a maximum of 55.0 points under relation-inversion prompts. Entity deletion shows a similar pattern, averaging 41.7 points with improvements of at least 35.2 points across all routes. Partial deletion yields smaller but consistent gains (22.6 points on average), reflecting the need to preserve non-sensitive answer content while suppressing sensitive spans. Since both methods use the same local objective and route prompts, these results provide controlled evidence that graph-guided support expansion improves robustness to diverse recovery routes.

4 Related Work

Early LLM unlearning studies adapted gradient ascent or retain-regularised objectives to suppress forget examples (Jang et al., 2023; Maini et al., 2024). More recent methods address known failure modes of direct ascent: NPO slows catastrophic collapse through a preference-inspired

Dataset	Method	Complete		Entity/Instance		Partial	
		Leak↓	PPL↓	Leak↓	PPL↓	Leak↓	PPL↓
TOFU	GA	97.13	24.91	97.75	51.13	99.75	61.24
	GA+KL	97.13	24.91	97.75	51.13	88.25	42.44
	DPO	99.88	**1.24**	100.00	**1.30**	100.00	2.94
	NPO-TR	91.50	4.79	88.63	9.12	95.13	10.75
	AltPO	97.50	1.58	97.19	**1.30**	100.00	9.02
	GRU	38.13†	79.11	65.31	59.09	86.25	24.42
	SELU	92.88	1.29	97.88	1.32	98.75	**1.27**
	Seed-Only	93.25	2.61	96.25	4.40	100.00	7.90
	GRAPHSU (Proposed)	<u>46.83</u>	3.27	<u>**54.60**</u>	4.94	<u>**81.67**</u>	1.91
PISTOL	GA	100.00	**1.02**	100.00	1.05	100.00	1.20
	GA+KL	100.00	**1.02**	99.39	1.06	99.34	1.20
	DPO	99.83	**1.02**	100.00	**1.02**	100.00	1.28
	NPO-TR	98.92	**1.02**	94.50	1.10	99.17	1.39
	AltPO	95.33	1.15	99.50	1.03	63.67	3.13
	GRU	99.87	1.04	98.70	1.06	98.76	1.37
	SELU	96.66	**1.02**	100.00	1.03	97.78	**1.01**
	Seed-Only	56.33	1.09	26.33	1.10	41.39	1.22
	GRAPHSU (Proposed)	<u>**6.83**</u>	1.05	<u>**4.67**</u>	1.10	<u>**29.17**</u>	1.19

Table 2: Main forget–retain comparison on GPT-2 Medium across TOFU and PISTOL. Leak denotes soft leakage (%); PPL denotes retain perplexity; lower is better for both. Bold marks the best value in each column, regardless of method. Underlining marks the lowest-leakage result satisfying the operational retention constraint PPL ≤ 10. † marks the lower-leakage TOFU complete-deletion result that violates this constraint.

loss (Zhang et al., 2024); AltPO combines negative and positive preference signals to avoid nonsensical forget responses (Mekala et al., 2025); GRU rectifies gradients to mitigate the unlearning–retention trade-off (Wang et al., 2025b); and SELU uses an energy-based LoRA formulation with straight-through estimators (Chaiyapattanaporn et al., 2026). These algorithms mainly improve the update rule. GRAPHSU is complementary: it determines the support routes, masks, and strengths to which an update rule is applied.

5 Conclusions

We presented GRAPHSU, a graph-guided controller that treats selective unlearning as a support-route control problem: instead of editing only the explicitly identified forget seeds, it builds a weighted multi-view support graph, propagates request-conditioned relevance from the seeds, and applies graded forgetting strengths to the high-risk neighbours. GRAPHSU can be paired with existing unlearning objectives without modifying them. Across TOFU and PISTOL with GPT-2 Medium and LLAMA-3.2-3B-INSTRUCT, GRAPHSU achieves the lowest utility-feasible soft leakage in all deletion settings, reducing leakage by up to 49.5 percentage points over a matched Seed-Only ablation while maintaining low retain perplexity. These results suggest that the deletion scope, not only the unlearning loss, is a key determinant of reliable knowledge removal.

Limitations

Unlike complete or entity deletion, partial deletion requires removing sensitive spans while preserving the surrounding answer content, making the task inherently more constrained. A well-posed partial-deletion instance requires an answer that contains at least two clearly separable facts, one sensitive and targeted for removal, and one safe and retained so that the sensitive span can be suppressed without disturbing the remainder. This is a property of the span-level deletion setting itself rather than of any particular method, and the residual leakage observed in this setting largely reflects this structural difficulty.

More broadly, soft leakage measures behavioural recoverability under a fixed family of recovery-route probes rather than certified representational erasure, so our results are best interpreted as evidence of substantially reduced reachability of the target knowledge. GRAPHSU additionally performs offline fact extraction and support-graph construction; this cost is incurred once per corpus and amortised across deletion requests (Appendix H).

Ethical Considerations

Selective unlearning can support privacy, copyright compliance, removal of stale information, and safer model maintenance. It can also create a false sense of compliance if evaluation is weak, or be misused to remove safety-relevant knowledge. We therefore present GRAPHSU as an approximate unlearning

technique that should be paired with governance, audit trails, and independent verification.

References

P. Chaiyapattanaporn, P. Stenetorp, and Y. Chen. 2026. Selu: Energy-based targeted unlearning in llms. OpenReview submission.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14389–14408.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, and 38 others. 2024. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Proceedings of the 41st International Conference on Machine Learning*.

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

Abhilash Mekala, Vahid Dorna, Shivani Dubey, Akshat Lalwani, Dawid Koleczek, Mukund Rungta, Sadid Hasan, and Elita Lobo. 2025. Alternate preference optimization for unlearning factual knowledge in large language models. In *Proceedings of COLING*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372.

OpenAI. 2025. Introducing gpt-4.1 in the api. `https://openai.com/index/gpt-4-1/`. Accessed: 2026-06-16.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. 2023. TRAK: Attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 27074–27113. PMLR.

Garima Pruthi, Frederick Liu, Mukund Sundararajan, and Satyen Kale. 2020. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, pages 19920–19930.

Xinchi Qiu, William F. Shen, Yihong Chen, Nicola Cancedda, Pontus Stenetorp, and Nicholas D. Lane. 2024. How data inter-connectivity shapes llms unlearning: A structural unlearning perspective. *arXiv preprint arXiv:2406.16810*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Hadi Reisizadeh, Jiajun Ruan, Yiwei Chen, Soumyadeep Pal, Sijia Liu, and Mingyi Hong. 2025. Leak@k: Unlearning does not make LLMs forget under probabilistic decoding. *arXiv preprint arXiv:2511.04934*.

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

Bozhong Tian, Xiaozhuan Liang, Siyuan Cheng, Qingbin Liu, Mengru Wang, Dianbo Sui, Xi Chen, Huajun Chen, and Ningyu Zhang. 2024. To forget or not? towards practical knowledge unlearning for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1524–1537, Miami, Florida, USA. Association for Computational Linguistics.

Wenyu Wang, Mengqi Zhang, Xiaotian Ye, Zhaochun Ren, Pengjie Ren, and Zhumin Chen. 2025a. UIPE: Enhancing LLM unlearning by removing knowledge related to forgetting targets. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 25212–25227, Suzhou, China. Association for Computational Linguistics.

Y. Wang, Q. Wang, F. Liu, W. Huang, Y. Du, X. Du, and B. Han. 2025b. Gru: Mitigating the trade-off between unlearning and retention for llms. In *Proceedings of the International Conference on Machine Learning*.

Rongzhe Wei, Peizhi Niu, Hans Hao-Hsun Hsu, Ruihan Wu, Haoteng Yin, Mohsen Ghassemi, Yifan Li, Vamsi K. Potluru, Eli Chien, Kamalika Chaudhuri, Olgica Milenkovic, and Pan Li. 2025. Do LLMs really forget? evaluating unlearning with knowledge correlation and confidence awareness. In *Advances in Neural Information Processing Systems*. NeurIPS 2025 poster.

Ruiqi Zhang, L. Lin, Y. Bai, and S. Mei. 2024. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15686–15702, Singapore. Association for Computational Linguistics.

A Experiment Results on LLAMA-3.2-3B-INSTRUCT

Larger-backbone evaluation. We evaluate GRAPHSU with LLAMA-3.2-3B-INSTRUCT on TOFU to examine whether graph-guided support expansion transfers beyond GPT-2 Medium. All methods start from the same supervised fine-tuned checkpoint and use the same deletion manifests, evaluation prompts, decoding settings, and checkpoint-selection criterion. Lower values are better for both soft leakage and retain perplexity.

Complete deletion.

Method	Leak↓	PPL↓
Seed-Only	93.67	**1.05**
GA	99.70	1.08
GA+KL	94.33	1.09
DPO	94.33	1.30
NPO-TR	91.66	1.06
AltPO	90.00	1.08
GRU	93.33	1.08
SELU	93.66	**1.05**
GRAPHSU	**13.33**	1.96

Entity/instance deletion.

Method	Leak↓	PPL↓
Seed-Only	25.11	1.04
GA	36.00	1.20
GA+KL	38.66	1.05
DPO	92.22	1.70
NPO-TR	33.33	1.09
AltPO	34.67	1.15
GRU	86.33	1.03
SELU	35.66	**1.02**
GRAPHSU	**15.53**	1.35

Partial deletion.

Method	Leak↓	PPL↓
Seed-Only	94.33	1.56
GA	94.36	1.05
GA+KL	89.00	1.16
DPO	96.36	**1.03**
NPO-TR	89.66	1.07
AltPO	88.67	1.09
GRU	93.67	1.06
SELU	94.66	1.04
GRAPHSU	**87.01**	1.07

The larger-backbone results preserve the central pattern observed in the controlled GPT-2 Medium experiments: expanding the edit scope beyond the explicit forget seeds is most beneficial when the target remains reachable through related examples or entity-centred support routes. Partial deletion remains more difficult because the method must suppress only a sensitive span while preserving the remainder of the answer.

Perplexity consistency check. Retain PPL is computed by teacher-forced scoring on the same frozen retain set for a given comparison. Prompt and padding positions are masked, answer-token negative log-likelihood (NLL) is summed over the evaluation set and divided by the number of scored answer tokens, and the aggregate mean NLL is exponentiated once. Consequently, PPL must be at least one. The DPO partial-deletion value previously shown as 0.03 was mean NLL rather than PPL; the corrected value is $\exp(0.03) \approx 1.03$. This correction does not change the leakage ranking because DPO soft leakage is 96.36%.

Applicability beyond QA-formatted data and causal language models. GPT-2 Medium and LLAMA-3.2-3B-INSTRUCT are autoregressive causal language models (CLMs), not QA-specific architectures; the QA pairs used here are the benchmarks' data representation. More generally, a GRAPHSU node can represent a document, passage, or conversation, with deletion masks identifying sensitive spans, while the same semantic, symbolic, and gradient support views define request-conditioned support routes for non-QA corpora.

B Pareto Trade-Off Plots

The plots below show the forget–retain trade-off for the three evaluated model–dataset settings. Lower-left is better; the shaded region marks the utility-feasible region, and the dashed line denotes retain PPL $= 10$ where shown. To preserve legibility in the two-column EMNLP format, each model–dataset plot is rendered as a separate full-width vector figure.

The PPL ≤ 10 threshold is used only as a fixed operational guardrail to prevent severe retention collapse from being interpreted as successful unlearning; it is not presented as a universal cutoff. We therefore also inspect the unfiltered Pareto frontier, retaining every reported operating point regardless of PPL.

Table 3: Unfiltered Pareto-optimal points on GPT-2 Medium TOFU. Each pair is (soft leakage %, retain PPL); lower is better on both axes.

Setting	Pareto-optimal operating points
Complete	DPO (99.88, 1.24); SELU (92.88, 1.29); GRAPHSU (46.83, 3.27); GRU (38.13, 79.11)
Entity/instance	AltPO (97.19, 1.30); Seed-Only (96.25, 4.40); GRAPHSU (54.60, 4.94)
Partial	SELU (98.75, 1.27); GRAPHSU (81.67, 1.91)

GRAPHSU is Pareto-optimal in all three deletion settings. It has the lowest leakage among the reported methods under every PPL cap from 5 through 50. With no retention cap, GRU attains

lower complete-deletion leakage (38.13%) but at retain PPL 79.11. NPO-TR is only slightly above the original guardrail for partial deletion (PPL 10.75), yet its soft leakage remains 95.13% compared with 81.67% for GRAPHSU. The guardrail should therefore be read as one practical operating region rather than as a filter that defines the underlying trade-off.

C Fact Extraction and GPT-4.1 Prompting

GRAPHSU uses GPT-4.1 only for offline fact grounding. For each QA pair, GPT-4.1 extracts schema-constrained head–relation–tail facts. The outputs are then validated, canonicalised, deduplicated, and type-checked using deterministic post-processing. Extraction is performed once per corpus and cached; it is not repeated for each deletion request. The recovery-route prompts in Appendix E are generated locally and do not use an external LLM.

Table 4: Fact-extraction settings.

Setting	Value
Extraction model	GPT-4.1
Output format	JSON object (`json_object` mode)
Nominal workload	Approximately one initial request per QA sample
Maximum retry attempts	3
Request timeout	60 seconds
Maximum labelled few-shot examples	20

System instruction.

```
You are an information extraction assistant that prepares
      data for a knowledge graph. Be CONSERVATIVE and
      ACCURATE: never invent books, awards, dates, or
      relationships that are not clearly stated in the
      question or answer. For each question/answer pair,
      extract ATOMIC factual triples suitable for graph edges.
       Each triple is an object with:
- "head": the entity the fact is about
- "head_type": one of [PERSON, ORG, WORK, EVENT, LOCATION,
      AWARD, GENRE, CHARACTER, TOPIC, OTHER]
- "relation": a short, descriptive edge label
- "tail": the target node, value, or linked entity
- "tail_type": one of [PERSON, ORG, WORK, EVENT, LOCATION,
      DATE, NUMBER, OCCUPATION, AWARD, GENRE, CHARACTER,
      TOPIC, NATIONALITY, LANGUAGE, OTHER]
BOOKS/WORKS: create a separate triple per titled work; use
      PERSON--author_of-->WORK; quoted strings in a works
      context are strong WORK candidates; add WORK--genre-->
      GENRE and WORK--publication_year-->DATE when stated.
      LANGUAGE/NATIONALITY: tag tail_type accordingly.
Use both question and answer; split comma/"and" lists into
      separate triples; do not invent facts. Return ONLY: {"
      triples": [...]}
```

Per-sample user prompt and worked example.

```
Study the supplied canonical examples, then extract graph
      triples for:
Question: Where does Avery Collins work?
Answer: Avery Collins works at Northbridge University and
      writes in Spanish.

-> {"triples": [
  {"head":"Avery Collins","head_type":"PERSON",
   "relation":"works_at","tail":"Northbridge University","
      tail_type":"ORG"},
  {"head":"Avery Collins","head_type":"PERSON",
   "relation":"writes_in","tail":"Spanish","tail_type":"
      LANGUAGE"}
]}
```

D Local Fact Extraction and Error Propagation

Hosted fact extraction may be undesirable for proprietary corpora. We therefore rebuilt the symbolic graph with the local Qwen2.5-3B-Instruct model and repeated a matched TOFU complete-deletion experiment. Only extractor-derived facts and symbolic edges changed; GPT-2 Medium, all 184 forget targets, training code, seed 42, unlearning recipe, evaluation manifest, decoding, and detector thresholds were held fixed.

Qwen processed 4,005 QA examples and returned valid triples for 3,942 (98.43%) without external API calls. In a 200-example structured audit, exact-triple F1 was 15.57% for Qwen and 9.88% for GPT-4.1; GPT-4.1 retained higher exact-tail accuracy. Because semantically equivalent facts can be decomposed into different triples, this structured audit is diagnostic rather than a complete semantic gold standard.

The two rebuilt graphs remained close despite extractor differences: edge-weight Spearman correlation was $\rho = 0.925$, degree correlation was $\rho = 0.979$, diffusion-score correlation was $\rho = 0.996$, and the selected support sets had Jaccard similarity 0.895 (68 of 72 selected supports were identical). To test error propagation, we additionally corrupted graph-active heads, tails, and triples. At 40% random-active corruption, 69 of 72 support nodes were retained; under 40% graph-critical corruption, 66 of 72 were retained. Diffusion-score correlations remained at least 0.9974. These perturbations indicate that symbolic noise is attenuated by the semantic and gradient views, weighted fusion, and diffusion rather than being automatically propagated to the full support closure.

At this matched operating point, local extraction gives soft leakage 49.83%, approximately 3.0 percentage points above the re-verified GPT-4.1-based GRAPHSU result (46.83%), while avoiding external transmission of the corpus. Leakage decreases by 36–44 percentage points across every tested route family and hard leakage is 0–3%. The trade-off is not cost-free: lexical retain utility falls, and neighbour-retain PPL reaches 11.54 even though the three-panel aggregate retain PPL

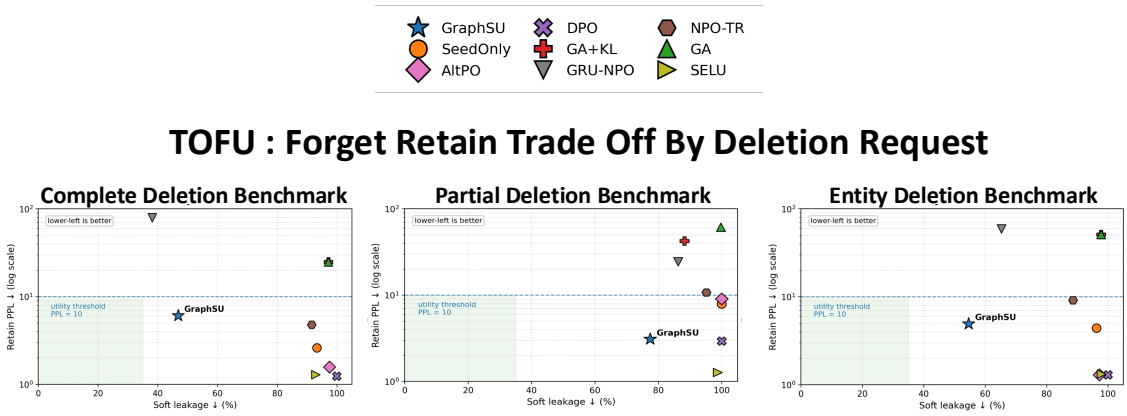

Figure 3: Forget–retain trade-off for GPT-2 Medium on TOFU. The complete, partial, and entity/instance deletion panels are shown at full width so axis labels, method labels, and the utility-feasible region remain readable.

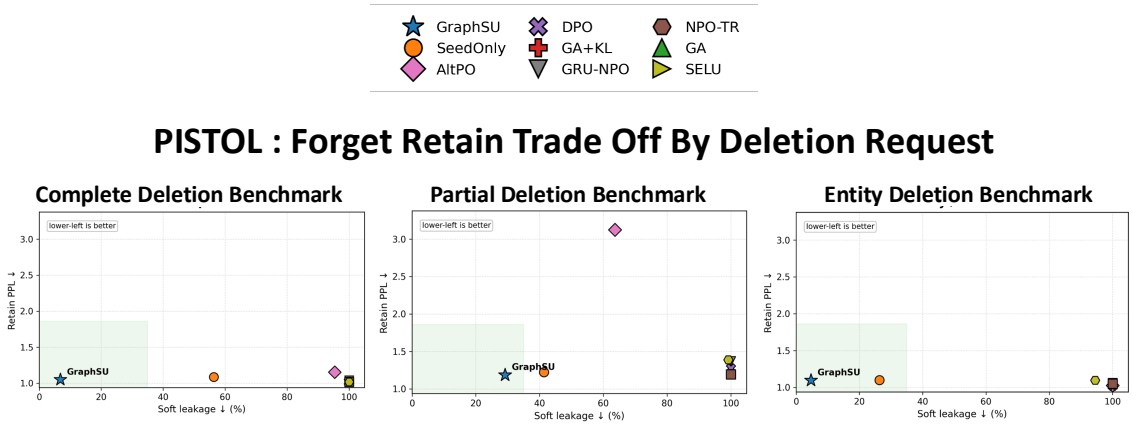

Figure 4: Forget–retain trade-off for GPT-2 Medium on PISTOL, shown as a full-width vector panel for improved readability.

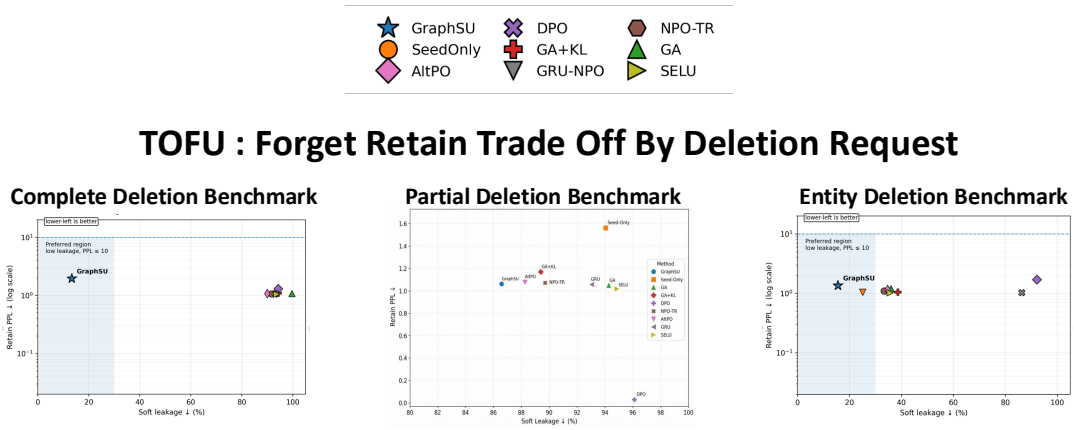

Figure 5: Forget–retain trade-off for LLAMA-3.2-3B-INSTRUCT on TOFU, shown as a full-width vector panel for improved readability.

is 9.95. We therefore view local extraction as a privacy-preserving deployment option with a measurable forgetting–utility trade-off, not as an accuracy-equivalent replacement for GPT-4.1.

E Adversarial Route-Prompt Generation

For each evaluated sample $z_i = (q_i, a_i^{\text{text}}, \mathcal{F}_i)$, we select a grounded target fact $k_i = (h_i, r_i, t_i)$, where h_i is the head entity, r_i is the relation, and t_i is the target value whose recoverability is assessed.

Table 5: Matched step-480 TOFU complete-deletion operating point using local Qwen2.5-3B-Instruct fact extraction. The Original-SFT column is the common pre-unlearning checkpoint.

Metric	Original-SFT	Local-Qwen GRAPHSU
Soft leakage ↓	91.25%	**49.83%**
Hard leakage ↓	37.67%	**1.33%**
Forget-answer F1 ↓	84.37%	**33.86%**
Retain F1 ↑	72.55%	30.27%
Retain semantic cosine ↑	90.95%	79.53%
Retain PPL ↓	4.30	9.95

Recovery-route prompts are generated offline using deterministic, relation-conditioned templates; no external LLM is used in this stage.

The generator may additionally use a validated alias set $A(h_i)$, an inverse-relation mapping $I(r_i)$, and related facts associated with the same entity or a graph-connected sample. Candidate prompts are normalised, deduplicated, checked for missing template slots, accidental target disclosure, and relation compatibility. The resulting manifests are frozen before evaluation and reused unchanged for GRAPHSU and all baselines. The task-specific monitoring panels use the prompt counts in Table 1; not every route is applicable to every target, and validated aliases may add entity-specific variants.

E.1 Detailed Worked Example

Consider the illustrative QA sample:

> **Question:** Where does Avery Collins work?
> **Answer:** Avery Collins works at Northbridge University.

The fact-extraction stage produces the grounded target fact with head *Avery Collins*, relation `works_at`, and tail *Northbridge University*. For this example, the generator is also provided with the validated aliases

$$A(h_i) = \{\text{A. Collins}, \text{Avery C.}\}, \tag{19}$$

and, when available, a related fact such as

$$(\text{Avery Collins}, \texttt{researches}, \text{machine ethics}). \tag{20}$$

All prompts below assess recovery of the same target value, Northbridge University; only the recovery route changes.

After template expansion, prompts pass through the following deterministic validation procedure:

1. fill route-specific slots using the grounded target fact, validated aliases, and available related facts;
2. remove prompts with missing or invalid slots;
3. remove prompts that reveal the target value, except when the route definition requires it, as in relation inversion;
4. remove exact and normalised duplicates;
5. retain only prompts compatible with the target relation;
6. order or sample valid variants using the fixed experimental seed; and
7. save the resulting manifest for reuse across all evaluated methods.

Thus, differences between GRAPHSU and the baselines cannot be attributed to different evaluation questions: each method uses the same frozen target facts, route prompts, decoding settings, and soft-leakage threshold.

F Taxonomy-Blind and Held-Out Leakage Probes

The templated route families in Appendix E are useful for controlled diagnosis, but robustness measured only on that taxonomy can overstate generalisation. We therefore perform an additional taxonomy-blind evaluation on all 184 TOFU complete-deletion targets. A local Qwen2.5-3B-Instruct model generates free-form probes using only the original question and answer. It receives no graph structure, route labels, templates, hyperparameters, or outputs from the unlearned models. The resulting probes have zero exact matches with the original template bank. Checkpoints and detector thresholds are frozen before this evaluation.

We evaluate 736 free-form probes (four per target) and a stronger attack panel of 342 probes across 171 targets, where each probe is first verified to recover the target answer from Original-SFT. Seed-Only and GRAPHSU use identical prompts and decoding. Leakage is measured with (i) a fixed-reference representation derived from Original-SFT, (ii) external MPNet semantic similarity, (iii) DeBERTa natural-language inference, (iv) direct target-phrase recovery, and (v) a conservative combined detector that fires when any constituent detector succeeds.

The direction is also consistent for direct phrase recovery, MPNet, and NLI considered separately. For continuity with the main evaluation, the original soft-leakage detector is retained as an auxiliary measure: on the full taxonomy-blind panel

Table 6: Recovery-route families and representative generation rules. The displayed prompts are illustrative; the implementation uses relation-conditioned template maps so that each prompt remains compatible with its grounded target relation.

Route family	Generation rule	Representative prompt
Direct	Preserve the original question, optionally adding a neutral instruction wrapper.	"Please answer: $\langle q_i \rangle$"
Paraphrase	Apply relation-specific lexical and syntactic transformations while preserving h_i, r_i, and target t_i.	"At which institution is $\langle h_i \rangle$ employed?"
Indirect	Ask for the same relation using contextual or descriptive wording rather than the original surface form.	"Which organisation is associated with $\langle h_i \rangle$'s employment?"
Cloze	Convert the target fact into a declarative statement and replace the target value with a blank.	"$\langle h_i \rangle$ works at ___."
Related-fact	Introduce another validated fact associated with h_i or a selected graph neighbour, and then query the target relation.	"The researcher who studies ⟨related topic⟩ is affiliated with which institution?"
Entity alias	Replace the canonical entity mention with a validated alias from $A(h_i)$.	"Where does $\langle \mathrm{alias}(h_i) \rangle$ work?"
List-summary	Request several profile attributes while ensuring that the target relation is included.	"List the main profile details for $\langle h_i \rangle$, including institutional affiliation."
Relation inversion	Reverse the grounded relation using $I(r_i)$ when the inverse is meaningful and sufficiently unambiguous.	"Which author is affiliated with $\langle t_i \rangle$?"

Table 7: Detailed route generation for one grounded target fact.

Route	Generated prompt	Purpose
Direct	Where does Avery Collins work?	Tests recovery through the original question.
Paraphrase	At which institution is Avery Collins employed?	Changes lexical and syntactic form while preserving the target relation.
Indirect	Which university is associated with Avery Collins's employment?	Avoids the original "where does ... work" formulation.
Cloze	Avery Collins works at ___.	Tests whether the target completes a factual statement.
Related-fact	The researcher known for work on machine ethics is affiliated with which institution?	Uses a related fact to reach the target entity; emitted only when that fact identifies the intended entity without ambiguity.
Entity alias	Where does A. Collins work?	Tests recovery after replacing the canonical entity name with a validated alias.
List-summary	List Avery Collins's main profile details, including institutional affiliation.	Tests whether the target appears within a broader profile response.
Relation inversion	Which author is affiliated with Northbridge University?	Reverses the relation direction; emitted only when the inverse mapping is defined and sufficiently unambiguous.

it favours GRAPHSU (93.48% versus 96.20% for Seed-Only; $p = 0.0187$), while it saturates on the smaller confirmed-attack panel. These results extend the claim from robustness within the predefined route family to improved robustness against unseen, independently generated recovery attempts; they do not constitute certified erasure.

G Qualitative Deletion Examples

The examples below use only the concepts defined in the main paper: the deletion setting, forget seed, grounded target fact, edit scope, support routes, and retention constraint. They illustrate how requests are represented and evaluated rather than introducing a separate response-transformation framework.

G.1 Complete deletion

Forget seed. *Question:* Where does Avery Collins work? *Answer:* Avery Collins works at Northbridge University. The grounded target has head *Avery Collins*, relation `works_at`, and tail *Northbridge University*.

Edit scope and interpretation. All answer tokens in the forget seed receive full forgetting pressure, while graph-selected support examples receive bounded pressure according to their request-conditioned support scores. A successful update prevents recovery of Northbridge University through applicable direct, paraphrase, cloze, related-fact, list-summary, and relation-inversion prompts while preserving utility on unrelated retain examples.

G.2 Entity/instance deletion

Deletion request. Remove the profile associated with Hsiao Yun-Hwa. One seed asks: *What is the full name of the author born in Taipei, Taiwan on 05/11/1991 who writes in the leadership genre?* The seed answer is: *The author's full name is Hsiao Yun-Hwa.*

Edit scope and interpretation. Canonical-name, alias, and entity-profile examples form the seed forget set, and additional high-risk support examples are selected through the request-conditioned graph closure. A successful update prevents recovery of the target profile through canonical-name, alias, indirect, related-fact, and list-summary prompts

Table 8: Taxonomy-blind leakage on unseen free-form probes for TOFU complete deletion. Reduction is Seed-Only minus GRAPHSU in percentage points. Confidence intervals are paired target-level 95% bootstrap intervals; p is the two-sided McNemar test.

Panel	Detector	Seed-Only	GRAPHSU	Reduction	95% CI	McNemar p
All targets	Fixed-reference	74.18%	**61.68%**	12.50 pp	[8.02, 17.26]	1.03×10^{-8}
All targets	Combined	75.68%	**62.09%**	13.59 pp	[8.97, 18.34]	5.98×10^{-10}
Confirmed attacks	Fixed-reference	78.95%	**65.20%**	13.74 pp	[7.02, 20.76]	4.92×10^{-5}
Confirmed attacks	Combined	79.82%	**65.50%**	14.33 pp	[7.89, 21.05]	2.23×10^{-5}

while preserving factual responses for unrelated entities. The target is suppressed without introducing a manually selected generic identity.

G.3 Partial deletion

Forget seed. *Question:* Where does Avery Collins work, and in which language do they write? *Answer:* Avery Collins works at Northbridge University and writes in Spanish. The sensitive target is the `works_at` value *Northbridge University*; the safe fact is that Avery Collins writes in Spanish.
Edit scope and interpretation. Forgetting is localised to the target span, while the safe answer span remains supervised; representation repulsion is disabled for this setting. A successful update prevents recovery of the institutional affiliation across applicable routes while retaining the model's ability to answer that Avery Collins writes in Spanish.

These examples clarify the three deletion granularities. Complete deletion suppresses the full answer associated with a seed; entity/instance deletion expands the request across an entity-centred profile and its aliases; and partial deletion suppresses only the grounded sensitive span while preserving the safe answer remainder.

G.4 Diagnosing Residual Leakage in the Support Graph

The support graph localises potential support at the *training-example level*; it does not claim to identify a single neuron or parameter in which a fact is stored. When a target remains recoverable after unlearning, the request-conditioned graph provides a diagnostic view through each node's diffusion score r_i, hop distance from the forget seed, and the semantic, symbolic, and gradient views contributing to its incident edges. Residual leakage can arise when relevant evidence lies just below the selection threshold or top-B expansion budget, when support is distributed across several individually weak paths, or when partial deletion intentionally limits forgetting pressure because sensitive and safe facts share the same answer context. Thus, difficult-to-remove information may correspond to redundant or diffuse support routes rather than a single missed example. This interpretation is diagnostic rather than a calibrated causal localisation claim.

H Operational Cost, Memory, and Scalability

Unless otherwise stated, the measurements below use GPT-2 Medium on TOFU. Corpus-level facts and graph artefacts are cached and reused across deletion requests.

H.1 Graph-construction scaling benchmark

We benchmark nested TOFU subsets using GPT-2 Medium. GPT-4.1 fact extractions were already cached and are therefore excluded from the build-time measurements. Embedding and symbolic processing ran on the CPU, while gradient-alignment features used the GPU. The graph degree is capped at 30.

At 4,000 samples, graph construction takes 452.36 seconds (about 7.5 minutes), cached artefacts occupy 7.82 MB, and warm support selection takes 4.01 seconds. Because the stored degree is capped at $K = 30$, the number of stored edges is $O(NK)$ and therefore linear in N for fixed K; graph storage and each sparse diffusion iteration scale with the stored edge set. The current exact cosine candidate-retrieval step is approximately quadratic in the number of samples and may become the dominant bottleneck at much larger corpus sizes. Approximate nearest-neighbour retrieval for the semantic view and symbolic inverted indices are natural scalable replacements that preserve the sparse downstream graph.

H.2 Request-time and audit costs

The isolated warm/cold timings in Table 9 measure graph loading and support selection on cached subset artefacts. The full request-conditioned pipeline additionally includes request grounding, one- and two-hop retrieval, support scoring, and retain-safety filtering.

Table 9: Graph construction, memory, cache size, and request-conditioned selection cost as the number of stored QA samples increases. Warm selection reuses in-memory cached artefacts; cold selection includes loading cached graph artefacts before support selection.

Samples	Stored edges	Avg. degree	Build time (s)	CPU RAM (GB)	GPU mem. (GB)	Cache (MB)	Warm sel. (s)	Cold sel. (s)
1K	11,279	22.56	106.01	1.41	2.36	2.87	0.68	1.27
2K	22,795	22.80	212.41	1.45	2.47	4.93	1.74	3.02
4K	45,524	22.76	452.36	1.40	2.36	7.82	4.01	6.53

Table 10: Stage-level operational costs from the full deletion workflow.

Stage	Observed cost or workload
Request-conditioned support selection	11.4–33.4 seconds per deletion request (mean 25.9 seconds), including request grounding, one- and two-hop support retrieval, support scoring, and retain-safety filtering. No external calls are used.
Model update and orchestration	Approximately 5.7–6.9 minutes per request, measured as total wall time minus logged evaluation time. The residual includes setup and checkpointing and is not a pure training-time measurement.
Full recovery-route audit	63.0–83.1 minutes and 4,464–5,376 local model generations. This is an offline validation cost rather than request-time latency and uses no external LLM calls.

With cached corpus artefacts, the recurring graph-specific overhead is small relative to model updating and the offline recovery-route audit. Fact extraction and graph construction are one-time corpus-level preprocessing costs and are amortised across deletion requests.

I Additional Utility Evaluation Beyond PPL

Perplexity is useful for detecting broad language-model degradation, but it does not fully capture whether retained factual behaviour is preserved. We therefore interpret soft leakage jointly with retain F1, ROUGE-L F1, far-retain F1, neighbour-retain F1, and safe-span F1 for partial deletion. The PPL ≤ 10 criterion used in the main paper is an operational guardrail for these experiments rather than a universal unlearning threshold.

Table 11: Primary TOFU forgetting and utility metrics for the controlled Seed-Only comparison.

Setting	Method	Soft leak↓	Retain PPL↓	Retain F1↑	ROUGE-L F1↑
Complete	Seed-Only	93.25	**2.61**	39.07	35.03
Complete	GRAPHSU	**46.83**	3.27	**42.41**	**47.36**
Entity	Seed-Only	96.25	**4.40**	37.16	33.38
Entity	GRAPHSU	**54.60**	4.94	**59.11**	**56.82**
Partial	Seed-Only	100.00	7.90	3.20	3.01
Partial	GRAPHSU	**81.67**	**1.91**	**55.14**	**51.07**

Table 12: Locality-sensitive retain metrics corresponding to Table 11. Safe-span F1 applies only to partial deletion.

Setting	Method	Far-retain F1↑	Neighbour F1↑	Safe-span F1↑
Complete	Seed-Only	41.05	31.79	–
Complete	GRAPHSU	**51.06**	**32.77**	–
Entity	Seed-Only	43.42	34.69	–
Entity	GRAPHSU	**57.13**	**46.17**	–
Partial	Seed-Only	1.14	0.83	1.26
Partial	GRAPHSU	**54.58**	**54.54**	**61.31**

The controlled comparison shows a stronger forgetting–utility balance for GRAPHSU in all three settings. Under complete deletion, GRU reaches lower soft leakage (38.13%) than GRAPHSU but raises retain PPL to 79.11, illustrating why leakage should not be interpreted without retention. Under partial deletion, NPO-TR has a slightly higher safe-span F1 (64.97) than GRAPHSU (61.31), but its retain PPL is 10.75 and soft leakage remains 95.13%. Neighbour-retain F1 is preserved or improved by GRAPHSU relative to Seed-Only: 32.77 vs. 31.79 for complete deletion, 46.17 vs. 34.69 for entity/instance deletion, and 54.54 vs. 0.83 for partial deletion.

J Ablation Coverage and Hyperparameter Scope

The controlled Seed-Only comparison is an ablation of the graph-guided scope controller as a whole: it uses the same local unlearning objective but omits graph-neighbour expansion. The local-extractor and corruption experiments in Appendix D further perturb the symbolic component while leaving the semantic and gradient views available. Together, these experiments establish that graph-guided support expansion matters and that the fused graph is robust to substantial symbolic noise.

The present experimental archive does not contain a complete leave-one-view-out sweep for all five affinity terms or independent performance sweeps over the fusion weights, restart probability ρ, hop limit, and expansion budget B. We there-

fore do not attribute the observed gains to any single graph view and do not claim that the reported hyperparameters are optimal. The evaluated configuration uses all five views, the fusion weights reported in Table 1, one- and two-hop support retrieval, and the task-specific neighbour caps in Table 1. A full factorial sensitivity analysis would require additional training runs and remains an important direction for establishing which views are necessary under different data regimes.

K Evaluation Consistency and Statistical Reliability

All reported retain PPL values use the same teacher-forced protocol within a comparison: prompt and padding positions are masked, answer-token NLL is accumulated over the frozen retain set, the total is divided by the number of scored answer tokens, and the resulting mean NLL is exponentiated once. This definition implies PPL ≥ 1 and motivated the correction in Appendix A. The local-extractor experiment additionally reports the Original-SFT retain PPL (4.30) at its matched operating point.

The unlearning runs reported in the paper use one training seed. We therefore do not describe evaluation resampling as across-seed variance. For the independently generated taxonomy-blind probes, Table 8 reports paired target-level bootstrap confidence intervals and two-sided McNemar tests because the same probes are evaluated under both checkpoints. The available aggregate benchmark outputs do not support a defensible reconstruction of across-seed variance for all baselines; accordingly, no such variance is fabricated. Training-seed sensitivity remains a limitation of the current empirical study. Future multi-seed comparisons should use paired target-level intervals for leakage, paired example-level intervals for retain F1/ROUGE-L, and example-resampled aggregate NLL before exponentiation for PPL, with multiplicity correction when several baselines are tested.