SCOPE BEFORE UPDATE: GRAPH-GUIDED PAIRED SUPERVISION FOR ENTITY UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models can reproduce sensitive or copyrighted information, motivating machine unlearning to remove selected knowledge while preserving other capabilities. Existing pipelines commonly require users to supply a forget set, which can re-expose sensitive content and complicate request auditing. We study a user-minimized interface in which the requester identifies a target entity and the service constructs the supervision needed for the update. This shifts the challenge to recovering useful target associations and separating behavior to change from nearby knowledge to preserve. We introduce AERIS (**A**ssociation-guided **E**ntity **R**emoval with **I**ntegrated **S**upervision), which organizes repeatedly elicited target-model associations into a weighted local graph and converts its paths into paired forget and neighbor data. The constructed supervision requires no original training passages and can be reused with standard unlearning objectives. Experiments on RWKU and TOFU show stronger forgetting than direct self-generation across the evaluated update objectives. With forget data fixed, graph-local neighbor supervision improves nearby knowledge preservation over Wikipedia retain data under both regularized objectives. On eight shared RWKU targets, AERIS reduces Forget-All by 20.2% relative to DirectQA under GA+GD.

1 INTRODUCTION

Large language models can reproduce sensitive or copyrighted information about people and other entities, motivating requests to remove selected knowledge while preserving the rest of the model's capabilities (Liu et al., 2025). Fine-tuning-based unlearning implements such requests using forget examples to specify the behavior to change and retain examples to constrain collateral effects. This training interface leaves an important deployment question about who should assemble the target-specific data needed to carry out a request.

Requiring requesters to supply that data introduces burdens beyond stating what should be forgotten. Users may have to collect and transmit sensitive content again, creating opportunities for secondary leakage (Jin et al., 2024). Providers must inspect the submitted content's authorization and scope, while malicious update data can induce backdoors, utility degradation, or reversible forgetting (Liu et al., 2024b; Arazzi et al., 2025; Ren et al., 2025b; Shang et al., 2025). These concerns motivate a **user-minimized request interface** in which the user supplies an anchor sufficient to identify the entity, such as a name with a short disambiguating description, and the service constructs the training examples (Figure 1). The requester identifies the entity without assembling its descriptions, relations, and question-answer examples into a training corpus. The provider then queries its own model for the examples used in the update.

Moving supervision construction to the service introduces two linked challenges. The first is **recovering target knowledge beyond direct recall**. A model may identify a person from a song title or a collaborator even when a direct request to describe that person omits those connections. Without original training passages or a user-supplied corpus, the service must find such contextual routes to the target rather than train only on the facts in one generated description. The second is **separating forgetting from nearby preservation**. Those same connections involve knowledge about other entities. Removing a singer's associations should not indiscriminately erase knowledge about their collaborators. A useful construction must therefore supply examples that change target behavior together with examples that preserve nearby non-target behavior.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

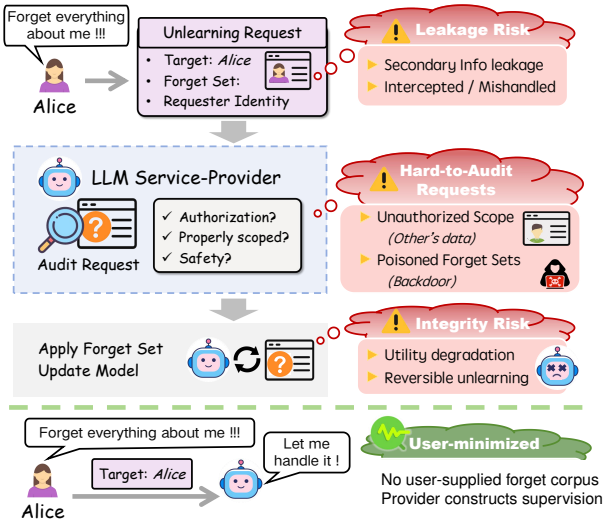


Figure 1: Providers turn minimal-anchor entity requests into forget and neighbor supervision.

Existing approaches obtain supervision from external text, auxiliary models, or target-model self-generation (Choi et al., 2025; Xie et al., 2025; Zhu et al., 2025; Ma et al., 2025). These sources make training examples available, but leave a design question about how to organize recovered associations for both forgetting and preservation. Our approach is to establish a common local association structure before the update. Paths through this structure provide contextual cues for target questions, while neighboring paths supply non-target preservation examples. We call this **scope before update**: selecting target-related contexts to train against and nearby non-target answers to preserve before changing model parameters.

We instantiate this principle in **AERIS** (**A**ssociation-guided **E**ntity **R**emoval with **I**ntegrated **S**upervision). Repeated, target-conditioned elicitation and frequency filtering assemble a weighted association graph. Coverage-aware paths and event-to-QA synthesis then produce paired forget and neighbor sets. The graph represents associations accessible under the elicitation protocol, with weights summarizing recurring cues. It connects the two construction challenges by using the recovered neighborhood to supply both training signals, which can be reused across update objectives.

We evaluate AERIS on RWKU's real-world entities (Jin et al., 2024) and TOFU's controlled author profiles (Maini et al., 2024). The results show stronger forgetting than direct self-generation across the evaluated objectives. With forget supervision fixed, graph-local neighbor data improve preservation over generic Wikipedia retain data under both regularized objectives. The recovered graphs also contain biographical attributes and neighboring entities that provide contexts for the synthesized questions. Our contributions are summarized below.

- A formulation of service-side supervision construction from minimal anchors, requiring neither user-supplied target corpora nor access to original training passages.
- A graph-guided method that connects target-conditioned association recovery to paired forgetting and nearby preservation supervision.
- Experiments across RWKU and TOFU demonstrate effective supervision across update objectives. On eight shared RWKU targets, AERIS lowers Forget-All by 20.2% relative to DirectQA under GA+GD; with forget data fixed, graph-local retain data raise Neighbor-Avg from 0.5644 to 0.5835 under the same objective.

2 Related Work

Unlearning objectives and supervision. LLM unlearning seeks to change selected model behavior while preserving other capabilities (Liu et al., 2025). Gradient ascent (GA) (Jang et al., 2023; Yao et al., 2024), preference-based objectives (Rafailov et al., 2023; Zhang et al., 2024; Ren et al., 2025a), and alternative-response training (Eldan & Russinovich, 2023) specify how to update a

Resource / step	RWKU	ELUDe	DirectQA	AERIS
External Data (Wiki)	✓	✓	✗	✗
Extra LLM / Human	✓	✓	✗	✗
Factual verification	✓	✗	✓	✗

Table 1: Resources and verification steps for target-specific supervision construction, excluding optional general retain data. ✓ denotes used and ✗ denotes not used.

model. Retain data, Kullback–Leibler (KL) divergence regularization, and knowledge replacement constrain these updates (Liu et al., 2022; Yuan et al., 2024; Xu et al., 2025). In-context methods, task vectors, and data sanitization offer additional approaches (Pawelczyk et al., 2023; Liu et al., 2024a; Bhaila et al., 2024). We construct forget and neighbor examples from target-model responses.

Constructing entity-level supervision. TOFU provides controlled fictional profiles (Maini et al., 2024), while RWKU evaluates real-world entities and neighboring knowledge (Jin et al., 2024). Entity-level studies examine forget-set quality, paraphrases, multi-hop questions, and locality (Ma et al., 2025; Yang et al., 2025; Chang & Lee, 2025). Related methods use concept-level interventions, external synthesis, or augmented forget data (Wang et al., 2025; Choi et al., 2025; Xu et al., 2025). Automated construction also uses domains, labels, seeds, and adversarial prompts (Zhu et al., 2025; Xie et al., 2025; Xu et al., 2026). Table 1 summarizes the resources and verification steps used to construct target-specific supervision, separately from optional general retain data. External synthesis relies on reference content and auxiliary annotation or generation, whereas DirectQA and AERIS query the target model itself. DirectQA is particularly relevant because it obtains self-verified QA from the target model. AERIS organizes repeated associations into a local graph and uses that same structure to construct both forget and neighbor data. Graph paths carry contextual cues into target-answer questions; paths from neighboring entities provide non-target answers for preservation. Both sets are constructed before parameter optimization and can be supplied to the same downstream unlearning objectives used with the other supervision sources.

Request interfaces and authorization. Malicious unlearning requests can induce backdoors, utility changes, or reversible forgetting (Liu et al., 2024b; Arazzi et al., 2025; Ren et al., 2025b; Shang et al., 2025), motivating careful request handling (Thaker et al., 2025). These risks concern the content entering an update as well as the update rule. Our user-minimized setting shifts target-specific example construction from the requester to the provider. For an accepted entity request, AERIS uses the target model to construct paired supervision. This makes supervision construction an explicit service-side stage connecting the request interface to an unlearning update.

3 Problem Setup

Let p_θ be the target language model and a_t an anchor identifying entity E_t. We assume that the service has resolved the entity identity, authorized the request, and specified its deletion scope. The requester supplies a_t, not target-specific training examples, and the service has no access to the original training passages. Let C denote the supervision construction procedure:

$$(S_f, S_n) = C(p_\theta, a_t), \qquad S_f = \{(x_i^f, y_i^f)\}_{i=1}^{m_f}, \quad S_n = \{(x_j^n, y_j^n)\}_{j=1}^{m_n}, \tag{1}$$

where S_f specifies target behavior to change and S_n provides nearby behavior to preserve. In our construction, x is a question and y its answer. An unlearning procedure U updates the model

$$\theta' = U(\theta, S_f, S_n, S_r), \tag{2}$$

where S_r is optional general-purpose retain data. We use *corpus-free* to denote target-specific supervision constructed from the model and entity anchor; optional general retain data are supplied through S_r. We study C while using established choices of U, evaluating $p_{\theta'}$ on target, neighbor, and general-utility probes. Benchmark entities and probes specify the experimental scope. The distinction between S_n and S_r is their scope: S_n contains questions about entities in the recovered target neighborhood, whereas S_r can cover broader non-target behavior. This keeps local preservation tied to the entity request while allowing general retention to enter the update separately.

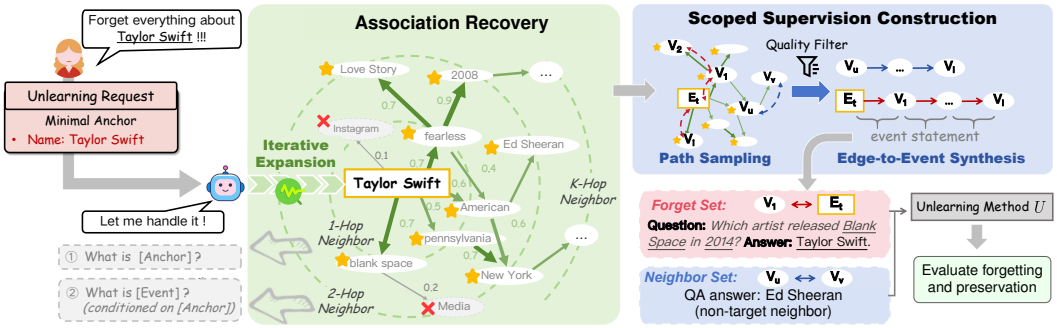


Figure 2: AERIS constructs paired supervision through target-conditioned expansion. Repeated elicitation builds a local association graph. Filtered paths provide contexts for target-answer and non-target neighbor supervision. Both sets feed an existing unlearning objective. Graph labels and weights are illustrative.

4 AERIS: Graph-Guided Paired Supervision

Given an accepted entity request, AERIS constructs the supervision that a user-supplied forget corpus would otherwise provide, together with local preservation examples. It connects two stages. **Association recovery** organizes repeated target-conditioned generations into a weighted graph, and **paired supervision construction** converts graph paths into S_f and S_n. Both stages query the target model before unlearning. Parameter optimization occurs downstream.

4.1 Recovering Target-Conditioned Associations

Entity knowledge can be elicited through different contextual cues. We expand those cues around E_t and consolidate repeated mentions into a local graph $G_t = (V_t, \mathcal{E}_t)$. Nodes represent extracted entities or events. Directed edges record associations generated in a source node's target-conditioned context. This *memory graph* represents associations elicited by the protocol.

To identify recurring associations, for each anchor context we collect N responses and measure the fraction containing candidate v as

$$s(v) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\big[v \in \text{Entities}(y_i)\big] \in [0, 1]. \tag{3}$$

Here y_i is one response. Entities canonicalizes mentions through case normalization and dictionary mappings, counting each distinct entity once per response. We retain candidates with $s(v) \geq \tau$. This frequency selects associations that recur across responses under the same anchor context and generation protocol. We use N=10, with τ=0.2 on RWKU and 0.3 on TOFU. Full settings appear in Appendix A.3.

We apply this filtering rule recursively. At hop zero, the graph contains E_t. After eliciting its neighbors, we expand each retained node u using a prompt conditioned on both E_t and u, up to K hops. This retains a common target context across the expansion. For a directed association (u, v), we normalize the co-elicitation count as

$$w(u, v) = \frac{c(u, v)}{\sum_{v' \in \mathcal{N}(u)} c(u, v')}. \tag{4}$$

The count $c(u, v)$ records the number of responses anchored at u that contain canonical entity v, counting it once per response; $\mathcal{N}(u)$ is the extracted neighborhood. Thus w is a within-context normalized association weight, distinct from the cross-response frequency s. The two normalizations serve different decisions: $s(v)$ determines whether a candidate recurs often enough to retain, whereas $w(u, v)$ compares the outgoing associations available for a sampling step. A candidate can therefore pass the recurrence threshold in several anchor contexts while receiving a different outgoing weight in each. The graph stores these directed weights for path sampling.

4.2 CONSTRUCTING PAIRED SUPERVISION

To turn the recovered associations into training contexts, we sample walks from E_t using edge weights and a visit penalty as

$$p(v \mid u, \mathcal{H}) \propto w(u, v) \left(\frac{1}{1 + \mathrm{vis}_{\mathcal{H}}(v)} \right)^{\alpha}, \tag{5}$$

where $\mathcal{H}$ is the sampling history, $\mathrm{vis}_{\mathcal{H}}(v)$ counts prior visits, and α controls exploration. Edge weights favor recurring associations, while the penalty increases opportunities to visit less-sampled nodes. We collect R walks with maximum length L and retain paths of at least two nodes satisfying

$$q(\pi) = \frac{1}{|\pi| - 1} \sum_{i=0}^{|\pi|-2} w(v_i, v_{i+1}) \geq \eta. \tag{6}$$

Edge weights guide each transition, while the visit history and path filter jointly determine the final sampling frequencies.

Target-answer supervision. For each retained path, a sliding window extracts adjacent pairs (v_i, v_{i+1}). We synthesize a short event statement linking each pair in the context of E_t and convert it into a QA example whose answer identifies E_t. For instance, "Which artist released Blank Space in 2014?" has the target answer Taylor Swift. Applying the forget loss to the answer span trains against target-identifying responses across the recovered contexts. A path of three nodes supplies two adjacent pairs, so a single walk can yield multiple target-answer contexts. Each question is generated from a local pair while retaining the target entity as its answer.

Neighbor supervision. We sample additional paths from strong neighbors while excluding E_t from the path, then synthesize QA with non-target neighbor answers. These examples form S_n, a local preservation constraint paired with S_f. The resulting questions retain the graph-local context while directing the preservation objective to non-target answers. Under retention-regularized updates, the forget loss is applied to target answers in S_f, while S_n preserves neighbor answers.

The pairing is defined by a shared recovered neighborhood: S_f asks for the target through its associated contexts, while S_n asks for non-target neighbors. For a given pre-update model and target entity, the graph and QA sets are constructed before optimization and can be reused with GA, negative preference optimization (NPO), and their retention-regularized variants. Appendix B provides the elicitation and statement-to-QA templates used in this construction.

5 EXPERIMENTS

The proposed construction connects an entity request to paired training data through a recovered association graph. This raises four questions: **RQ1** Can these data remove target knowledge while preserving other behavior? **RQ2** How do local neighbor supervision and forget-set composition affect unlearning? **RQ3** Which target associations are recovered, and what contextual knowledge do they provide for supervision? **RQ4** How stable are the recovered associations under construction perturbations, and how can they be reused across update objectives?

5.1 EXPERIMENTAL SETUP

Benchmarks and models. **RWKU** (Jin et al., 2024) evaluates real-world public figures. Our four-source comparison uses the eight targets shared with ELUDe. A separate comparison with RWKU supervision covers all 200 entities. **TOFU** (Maini et al., 2024) uses fictional author profiles introduced by fine-tuning. We use `forget01`. RWKU runs use LLaMA-2-7B-Chat (Touvron et al., 2023) and Phi-3-Mini-4K-Instruct (Abdin et al., 2024). TOFU uses benchmark-finetuned LLaMA-2-7B-Chat and LLaMA-3.2-1B-Instruct (Meta, 2024) checkpoints.

Supervision sources and update objectives. We compare four supervision sources: benchmark supervision, **ELUDe**'s external Wikipedia/GPT-4o synthesis (Choi et al., 2025), **DirectQA**'s target-model self-generation and self-verification (Ma et al., 2025), and **AERIS**. Table 1 summarizes their

Finetune	Method	Forget Set ↓				Neighbor Set ↑			MIA Set		Utility Set ↑				
		FB	QA	AA	All	FB	QA	All	FM ↑	RM ↓	MMLU	BBH	TruthfulQA	TriviaQA	AlpacaEval
Before		0.6788	0.6721	0.6829	0.6779	0.6306	0.7484	0.6895	1.9104	2.0433	0.4240	0.2654	0.2875	0.4056	6.2993
GA	RWKU	**0.5817**	0.5511	0.6536	0.5955	0.5600	**0.7153**	0.6376	**3.1075**	2.4128	0.3969	0.2346	0.3025	0.3675	6.3109
	ELUDe	0.6440	0.4646	0.6026	0.5704	0.6135	0.6944	0.6540	1.9933	2.1057	0.4167	0.2438	**0.3175**	0.3232	6.3341
	DirectQA	0.6531	0.6675	0.7030	0.6745	**0.6353**	0.6979	**0.6666**	1.9286	**2.0572**	**0.4203**	0.1975	0.2900	**0.4640**	6.2686
	AERIS	0.6373	**0.4288**	**0.5438**	**0.5366**	0.6263	0.6501	0.6382	1.9885	2.1016	0.4130	**0.2515**	0.3050	0.2955	**6.3446**
NPO	RWKU	0.6444	0.6128	0.6631	0.6401	0.6103	0.6925	0.6514	2.1667	2.2093	0.4137	0.2022	0.2925	0.4053	6.3267
	ELUDe	0.4290	0.2718	0.4061	0.3690	0.4829	0.4239	0.4534	**2.1874**	2.2383	0.3977	0.1867	**0.3525**	0.1526	**6.3930**
	DirectQA	0.6539	0.6842	0.7139	0.6840	**0.6364**	**0.7213**	**0.6788**	1.9178	**2.0494**	**0.4211**	0.2099	0.2900	**0.4658**	6.2765
	AERIS	**0.4177**	**0.1978**	**0.3734**	**0.3296**	0.4279	0.3800	0.4040	2.1054	2.1770	0.4123	**0.2438**	0.3300	0.1656	6.3849
GA+GD	RWKU	**0.5108**	0.4583	0.6075	0.5255	0.5691	**0.7214**	0.6453	**7.1499**	2.4194	0.3984	0.2191	0.3050	0.3708	6.2991
	ELUDe	0.5859	0.4006	0.5257	0.5041	0.5750	0.5864	0.5807	1.9600	2.0716	0.4152	0.1466	**0.3350**	0.2820	6.3393
	DirectQA	0.5468	0.6038	0.6772	0.6092	**0.6464**	0.6953	**0.6709**	1.8484	**1.9758**	**0.4174**	**0.2809**	0.2775	**0.4910**	6.2556
	AERIS	0.5928	**0.3644**	**0.5012**	**0.4861**	0.5882	0.5788	0.5835	2.0119	2.1121	0.4159	0.2741	0.3050	0.2469	**6.3499**
GA+KL	RWKU	**0.5108**	0.4644	0.6128	**0.5293**	0.5650	**0.7222**	0.6436	**7.1544**	2.4340	0.4050	0.2160	0.3050	0.3718	6.3051
	ELUDe	0.6170	0.4475	0.5797	0.5480	0.6148	0.6493	0.6320	1.9768	2.0909	0.4167	0.1003	**0.3175**	0.3151	6.3354
	DirectQA	0.5812	0.6152	0.6968	0.6311	**0.6239**	0.7064	**0.6651**	1.9559	**2.0635**	0.4152	0.2083	0.2850	**0.4517**	6.2774
	AERIS	0.6319	**0.4364**	**0.5538**	0.5407	0.6019	0.6295	0.6157	2.0046	2.1068	**0.4196**	**0.2358**	0.3000	0.3291	**6.3436**

Table 2: Unlearning performance with different fine-tuning strategies and forget-set generation methods on LLaMA-2-7B-Chat, averaged over eight shared RWKU/ELUDe targets. Bold denotes the best result and underlined denotes the second best. Arrows indicate the direction of preference (↓ lower is better; ↑ higher is better).

construction resources and verification steps. DirectQA generates entity-specific QA candidates and retains pairs that pass five rounds of self-verification, following Ma et al. (2025). Candidate generation continues to the data-scale target set by the reference supervision; DirectQA uses Wikipedia retain data for regularized updates. Appendix A.2 describes this construction protocol. We use gradient ascent (**GA**), negative preference optimization (**NPO**), and two retention-regularized objectives: **GA+GD** combines GA with gradient descent (GD) on retain data, while **GA+KL** combines GA with KL divergence regularization on retain data. Learning rates are fixed within each objective, with epochs adjusted for approximate optimization-step matching (Appendix A.1). The RWKU AERIS GA+GD/KL configurations use graph-local neighbor data. Table 3 separately varies the retain source with forget data fixed.

Evaluation metrics. For RWKU, **FB**, **QA**, and **AA** denote fill-in-the-blank, direct question-answer, and adversarial-attack probes. They use **ROUGE-L recall**, with **Forget**↓ and **Neighbor**↑. **All** aggregates each group's probes. **FM/RM** are forget-/retain-member scores from the membership-inference evaluation, with the reported directions FM↑/RM↓. **Utility** is reported per task. For TOFU, **Forget Q.** is a KS-test p-value comparing truth-ratio distributions with a retain-only reference. **Model Utility** aggregates nine non-forget metrics by harmonic mean. Higher is better for both. Appendix A.4 defines all evaluation metrics. Implementation, construction hyperparameters, and objective definitions appear in Appendices A.1, A.3, and A.5. Appendix D.8 identifies the TOFU checkpoints.

5.2 RQ1. Does the constructed supervision enable forgetting?

Forgetting across update objectives. We test whether service-constructed examples supply an effective forgetting signal across update objectives. The controlled RWKU comparison uses the eight targets shared by all four supervision sources (Table 2). AERIS improves target forgetting over DirectQA under every objective and achieves the lowest Forget-All under GA, NPO, and GA+GD. The gains also appear in direct and adversarial queries: under GA+GD, QA recall decreases from DirectQA's 0.6038 to 0.3644 and AA recall from 0.6772 to 0.5012. The same construction supports distinct update objectives: NPO reaches Forget-All 0.3296 and Neighbor-All 0.4040; GA+KL reaches Forget-All 0.5407 and Neighbor-All 0.6157 with preservation regularization.

Across all 200 RWKU entities (Table 4), AERIS improves forgetting over benchmark supervision under every objective, with comparatively small changes in neighbor retention. This comparison evaluates the constructed supervision at benchmark scale.

TOFU provides a complementary test on controlled fictional profiles (Table 5). Under GA, AERIS raises Forget Q. from DirectQA's 0.0286 to 0.0541, with Model Utility of 0.5155 compared with 0.5783. Under NPO, Forget Q. increases from 0.0541 to 0.0971, with Model Utility of 0.3949 compared with 0.5743. Additional objectives and model families are reported in Appendices C.1 and C.2.

Objective	Retain source	Forget-Avg ↓	Neighbor-Avg ↑
GA+GD	Wiki	**0.4834**	0.5644
	AERIS neighbor	0.4861	**0.5835**
GA+KL	Wiki	**0.5353**	0.6030
	AERIS neighbor	0.5407	**0.6157**

Table 3: Retain-source ablation with fixed forget data.

Objective	Source	Forget All ↓	Neighbor All ↑
Before		0.6742	0.7492
GA	RWKU	0.6682	0.7390
	AERIS	**0.6557**	0.7455
NPO	RWKU	0.6640	0.7366
	AERIS	**0.6346**	0.7330
GA+GD	RWKU	0.6614	0.7368
	AERIS	**0.6552**	0.7390
GA+KL	RWKU	0.6588	0.7357
	AERIS	**0.6556**	0.7337

Table 4: RWKU over all 200 entities, measured by ROUGE-L recall.

Objective	Source	Forget Q.↑	Model Utility↑
Before		0.0013	0.6277
GA	TOFU	**0.1650**	0.3309
	DirectQA	0.0286	**0.5783**
	AERIS	0.0541	0.5155
NPO	TOFU	**0.1650**	0.2951
	DirectQA	0.0541	**0.5743**
	AERIS	0.0971	0.3949

Table 5: TOFU on LLaMA-2-7B-Chat. Forget Q. is a KS-test p-value; Model Utility is the harmonic mean of non-forget metrics. Higher is better for both.

5.3 RQ2. How do local retention and forget-set composition shape unlearning?

We examine the paired supervision through two complementary analyses: varying the retain source with forget data fixed, and varying association coverage or factual composition within the forget set.

Local neighbors improve preservation. Replacing generic Wiki retain data with AERIS's graph-local neighbors raises Neighbor-Avg from 0.5644 to 0.5835 under GA+GD and from 0.6030 to 0.6157 under GA+KL (Table 3). Forget-Avg changes from 0.4834 to 0.4861 and from 0.5353 to 0.5407. The Neighbor-Avg gains are 0.0191 and 0.0127, respectively, while Forget-Avg changes by 0.0027 and 0.0054. This comparison isolates the retain source by using the same forget examples for both settings.

Coverage and factual composition. Figure 4 plots forgetting and neighbor retention at different graph-coverage levels. We hold QA-set size approximately constant and average coverage across targets. Forget-All varies more across these settings, while Neighbor-All remains within a narrower range. Under GA, Figure 3 varies the fraction of GPT-4-labeled factually correct QA from 0% to 100%, with comparatively small changes in the displayed performance curves (Felkner et al., 2024). Coverage changes the range of graph entities represented in a similarly sized QA set. Correctness mixtures change the proportion labeled factually correct; the forget loss still acts on the target answers. Appendices D.2 and D.4 provide the corresponding plots, and Table 10 reports the full retain-source comparison.

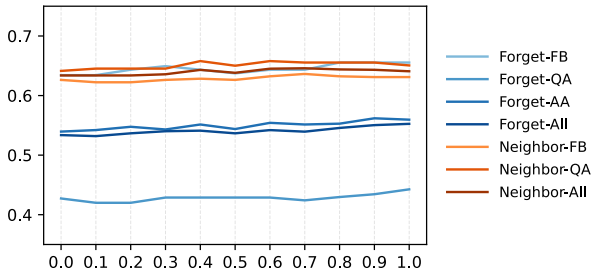


Figure 3: Correct QA fraction and ROUGE-L recall.

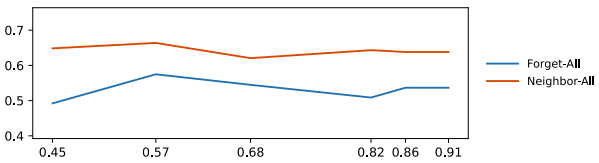


Figure 4: Graph coverage and ROUGE-L recall.

Attributes	Freq	DirectQA		AERIS	
		Covered	F1	Covered	F1
BirthPlace	2	✓	1.00	✓	0.80
Gender	1		0		0
BirthYear	1		0	✓	0.67
Genre	3	✓	0	✓	0.67
Awards	1		0	✓	1.00
Par_Job	4	✓	0.80		0
Books	3		0	✓	0.80
Books	1		0	✓	0.33

Table 7: Comparison of recovered attributes against TOFU ground-truth profiles.

Metric	Value
Density	0.01±0.00
Gini-In	0.64±0.04
Gini-Out	0.76±0.03
MaxDeg-In	98.16±36.08
MaxDeg-Out	24.74±3.88
Trans.	0.19±0.05
#Comm.	11.75±2.91
Modularity	0.41±0.07

Table 8: Recovered graph structure: mean ± standard deviation over 200 RWKU targets.

5.4 RQ3. What contextual knowledge does association recovery provide?

Source	Entity Cov. ↑	Wiki Cov. ↑
ELUDe	0.20±0.06	**0.52±0.13**
AERIS-Graph	**0.37±0.08**	0.32±0.12
AERIS-Response	0.19±0.05	0.45±0.13

Table 6: Wikidata one-hop recognizability (mean ± standard deviation across entities).

Knowledge alignment. On RWKU, the entity frequencies from AERIS and the external synthesis method ELUDe have cosine similarity 0.86. Against Wikidata, the proportion of model-side entities matched to the reference increases from 0.20 for ELUDe to 0.37 for AERIS-Graph. The fraction of Wikidata neighbors covered increases from 0.32 for the raw graph to 0.45 after synthesis. Entity Coverage divides the number of matched entities by the size of the generated entity set. Wiki Coverage divides the number of reached reference entities by the size of the Wikidata neighborhood. The denominators therefore distinguish the composition of the generated set from its coverage of the reference. The controlled TOFU example in Table 7 shows which profile attributes enter the recovered contexts. For Basil Mahfouz Al-Kuwaiti, AERIS covers birth-year, award, and book attributes absent from the DirectQA recovery in this example. These attributes supply distinct cues for questions about the same author. Appendix D.1 gives the answer-level example. Figure 6 and Appendix D.6 detail the pairwise and external comparisons.

5.5 RQ4. How stable and reusable is the recovered structure?

We measure graph topology across 200 RWKU entities and compare the associations elicited from different models. We then perturb the construction settings to measure changes in neighbor rankings and graph membership. Recovery costs and the conditions for reusing a constructed graph are reported at the end of this section.

Sparse neighborhoods and model-specific associations. Table 8 shows sparse graphs (density 0.01) with concentrated connectivity (in/out-degree Gini 0.64/0.76), local closure (transitivity 0.19), and community structure (modularity 0.41). The degree statistics summarize how edges concentrate on a subset of nodes; transitivity and modularity summarize local closure and community structure. The path sampler uses outgoing edge weights together with a visit penalty to explore this neighborhood. Figure 5 illustrates the corresponding semantic variation. American, the birth date, and Pennsylvania occur in all three models' top-eight lists for Taylor Swift. Other associations vary: GPT-4o includes Fearless, while LLaMA-3-8B-Instruct ranks Taylor Alison Swift first and includes Scott Borchetta. The lists combine identity attributes, locations, works, and associated people in different proportions. These shared anchors and distinct neighborhoods motivate constructing supervision for the model being updated.

Ranking and membership stability. Table 9 reports shared-neighbor rank correlations of 0.77–0.85 across prompt, seed, temperature, and threshold perturbations. Root-neighbor Jaccard is 0.82 under seed changes, 0.75 under prompt changes, and 0.66 at temperature 0.9. At the stricter threshold τ=0.6, shared-neighbor rank correlation is 0.84, with head-node and two-hop Jaccard scores of 0.23 and 0.26. Root-neighbor Jaccard is 0.62 under the same threshold change. These measures use different node sets: immediate neighbors, high-importance nodes, and the two-hop neighborhood. Rank correlation is computed on shared neighbors and remains high even when the retained sets differ. Thus the recurring neighbors have similar relative ordering across the tested settings, while

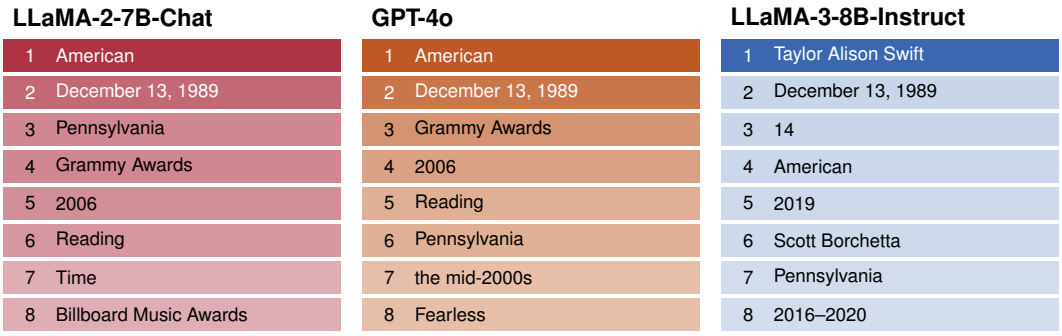


Figure 5: Top-eight associations elicited for Taylor Swift, ordered by rank within each model. Darker shading denotes higher association scores, scaled to the maximum within each model. Hues identify models. Appendix D.3 gives the expanded association lists.

	Ranking	Set membership (Jaccard)			Communities
Setting	Shared ρ	Root nbr.	Head nodes	2-hop ego	NMI
Prompt	0.85±0.09	0.75±0.08	0.49±0.04	0.48±0.04	0.48±0.06
Seed	0.82±0.19	0.82±0.04	0.50±0.04	0.49±0.05	0.48±0.06
Temp. (0.9)	0.77±0.17	0.66±0.11	0.46±0.05	0.42±0.06	0.43±0.05
Threshold (τ=0.6)	0.84±0.12	0.62±0.11	0.23±0.07	0.26±0.07	0.59±0.12

Table 9: Rank, membership, and community stability (mean ± standard deviation across entities). Definitions are in Appendix D.5.

graph membership depends on the construction choices. Appendix D.5 gives all five metrics and their definitions. Appendix D.7 complements these structural measurements with an exploratory analysis of elicitation behavior for entities with pageviews 20–50.

Recovery and reuse. Recovery averages 88.36 elicitation iterations per RWKU entity and 13.75 per TOFU entity; each iteration repeats one anchor-conditioned sub-query. These averages correspond to the two-hop RWKU and three-hop TOFU configurations. Each retained node can introduce another anchor context for expansion, with N=10 responses collected per context in both settings. Hop limits specify the expansion depth; the reported iteration counts summarize the resulting elicitation workload. For a fixed pre-update model and entity, this recovery is performed once. The resulting graph and paired QA sets can be cached for reuse across GA, NPO, and their regularized variants, separating supervision construction from each subsequent optimization run. Changing from GA to NPO changes how the target answers are optimized while the constructed examples remain fixed. Retention-regularized updates additionally use the neighbor answers in S_n to constrain changes to nearby knowledge.

6 Conclusion

AERIS turns a minimal entity request into a target-conditioned association graph and paired forget and neighbor supervision, moving training-data construction from the requester to the service provider. Repeated elicitation recovers recurring associations, and path-based synthesis converts them into target-focused and local preservation examples for existing unlearning objectives. Across RWKU and TOFU, this supervision improves benchmark forgetting over direct QA generation; with forget data fixed, graph-local neighbors improve preservation over generic Wiki retain data. On the eight shared RWKU targets, the GA+GD comparison yields a 20.2% relative reduction in Forget-All against DirectQA; replacing Wiki retain data with graph-local neighbors raises Neighbor-Avg from 0.5644 to 0.5835 with forget data fixed. For a fixed pre-update model and entity, the resulting data can serve multiple update objectives. This places the construction of target and preservation examples with the provider, allowing requests to identify an entity without supplying a training corpus.

AI USE STATEMENT

We used generative AI tools to polish the language of this manuscript and to generate training data as part of the proposed method. Specifically, the target language model produces the association statements and question-answer examples used to construct the forget and neighbor supervision, following the procedure and prompts described in the paper. The authors take full responsibility for the final manuscript, including its scientific claims, reported results, and AI-assisted content.

REPRODUCIBILITY STATEMENT

Appendix A documents the model configurations, data preprocessing, fine-tuning settings, DirectQA construction protocol, method hyperparameters, evaluation metrics, and unlearning objectives. Appendix B provides the prompts used for association recovery and supervision construction. Appendix C reports additional unlearning results across model families and update objectives. Appendix D details the recovery examples, correctness mixtures, coverage experiments, cross-model comparisons, graph stability tests, and Wikidata alignment analysis, together with qualitative observations for low-pageview entities. The TOFU checkpoint identifiers are listed in Appendix D.8.

REFERENCES

Marah I. Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. doi: 10.48550/arXiv.2404.14219.

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

Karuna Bhaila, Minh-Hao Van, and Xintao Wu. Soft prompting for unlearning in large language models. *arXiv preprint arXiv:2406.12038*, 2024.

Hwan Chang and Hwanhee Lee. Which retain set matters for LLM unlearning? a case study on entity unlearning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5966–5982, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.310. URL https://aclanthology.org/2025.findings-acl.310/.

Minseok Choi, Daniel Rim, Dohyun Lee, and Jaegul Choo. Opt-out: Investigating entity-level unlearning for large language models via optimal transport. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28280–28297, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. URL https://aclanthology.org/2025.acl-long.1371/.

Vineeth Dorna, Anmol Reddy Mekala, Wenlong Zhao, Andrew McCallum, J. Zico Kolter, Zachary Chase Lipton, and Pratyush Maini. OpenUnlearning: Accelerating LLM unlearning via unified benchmarking of methods and metrics. In *Advances in Neural Information Processing Systems: Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=Gy67Zh5X1i.

R Eldan and M Russinovich. Who's harry potter? approximate unlearning in llms, arxiv. *arXiv preprint arXiv:2310.02238*, 2023.

Virginia Felkner, Jennifer Thompson, and Jonathan May. GPT is not an annotator: The necessity of human annotation in fairness benchmark construction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14104–14115, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.760. URL https://aclanthology.org/2024.acl-long.760/.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 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), pp. 14389–14408, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.805. URL https://aclanthology.org/2023.acl-long.805/.

Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. Rwku: Benchmarking real-world knowledge unlearning for large language models. *Advances in Neural Information Processing Systems*, 37:98213–98263, 2024.

Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pp. 243–254. PMLR, 2022.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pp. 1–14, 2025.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Towards safer large language models through machine unlearning. *arXiv preprint arXiv:2402.10058*, 2024a.

Zihao Liu, Tianhao Wang, Mengdi Huai, and Chenglin Miao. Backdoor attacks via machine unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14115–14123, 2024b.

Weitao Ma, Xiaocheng Feng, Weihong Zhong, Lei Huang, Yangfan Ye, Xiachong Feng, and Bing Qin. Unveiling entity-level unlearning for large language models: A comprehensive analysis. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5345–5363, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.358/.

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

Meta. Llama 3.2 model card. https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct, 2024.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few shot unlearners. *arXiv preprint arXiv:2310.07579*, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Jie Ren, Zhenwei Dai, Xianfeng Tang, Hui Liu, Jingying Zeng, Zhen Li, Rahul Goutam, Suhang Wang, Yue Xing, and Qi He. A general framework to enhance fine-tuning-based llm unlearning. *arXiv preprint arXiv:2502.17823*, 2025a.

Jie Ren, Zhenwei Dai, Xianfeng Tang, Yue Xing, Shenglai Zeng, Hui Liu, Jingying Zeng, Qiankun Peng, Samarth Varshney, Suhang Wang, et al. Keeping an eye on llm unlearning: The hidden risk and remedy. *arXiv preprint arXiv:2506.00359*, 2025b.

Bingqi Shang, Yiwei Chen, Yihua Zhang, Bingquan Shen, and Sijia Liu. Forgetting to forget: Attention sink as a gateway for backdooring llm unlearning. *arXiv preprint arXiv:2510.17021*, 2025.

Pratiksha Thaker, Shengyuan Hu, Neil Kale, Yash Maurya, Zhiwei Steven Wu, and Virginia Smith. Position: Llm unlearning benchmarks are weak measures of progress. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 520–533. IEEE, 2025.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Huazheng Wang, Yongcheng Jing, Haifeng Sun, Yingjie Wang, Jingyu Wang, Jianxin Liao, and Dacheng Tao. Erasing without remembering: Implicit knowledge forgetting in large language models. *arXiv preprint arXiv:2502.19982*, 2025.

Linxi Xie, Xin Teng, Shichang Ke, Hongyi Wen, and Shenji Wan. Reveal and release: Iterative LLM unlearning with self-generated data. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 23887–23899, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.1298. URL https://aclanthology.org/2025.findings-emnlp.1298/.

Haoming Xu, Ningyuan Zhao, Liming Yang, Sendong Zhao, Shumin Deng, Mengru Wang, Bryan Hooi, Nay Oo, Huajun Chen, and Ningyu Zhang. Relearn: Unlearning via learning for large language models. *arXiv preprint arXiv:2502.11190*, 2025.

Xiaoyu Xu, Minxin Du, Zitong Li, Zi Liang, Zhibiao Guo, Shiyu Zhang, Peizhao Hu, Qingqing Ye, and Haibo Hu. From domains to instances: Dual-granularity data synthesis for LLM unlearning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 28540–28556, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1424. URL https://aclanthology.org/2026.findings-acl.1424/.

Nakyeong Yang, Minsung Kim, Seunghyun Yoon, Joongbo Shin, and Kyomin Jung. FaithUn: Toward faithful forgetting in language models by investigating the interconnectedness of knowledge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12999–13014, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.657. URL https://aclanthology.org/2025.emnlp-main.657/.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2024.

Xiaojian Yuan, Tianyu Pang, Chao Du, Kejiang Chen, Weiming Zhang, and Min Lin. A closer look at machine unlearning for large language models. *arXiv preprint arXiv:2410.08109*, 2024.

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

Xiaoyuan Zhu, Muru Zhang, Ollie Liu, Robin Jia, and Willie Neiswanger. LLM unlearning without an expert curated dataset. In *Proceedings of the 2nd Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=m4F3kQCfGX.

		Forget-Avg ↓	Neighbor-Avg ↑
GA+GD	Wiki	**0.4834**	0.5644
	Neighbor	0.4861	**0.5835**
GA+KL	Wiki	**0.5353**	0.6030
	Neighbor	0.5407	**0.6157**

Table 10: Impact of the Neighbor Set on Unlearning Performance

A Experiment Settings

A.1 Implementation Details

Models. We run RWKU experiments on the base LLaMA-2-7B-Chat (Touvron et al., 2023) and Phi-3-Mini-4K-Instruct (Abdin et al., 2024) models. For TOFU, we use the benchmark-specific LLaMA-2-7B-Chat and LLaMA-3.2-1B-Instruct checkpoints released by the TOFU and OpenUnlearning projects (Dorna et al., 2025), respectively, since the benchmark requires its injected author profiles to be present in the model. Exact artifact identifiers are listed at the end of this supplement.

Data preprocessing. We use released TOFU and RWKU splits and probes for evaluation. For AERIS and DirectQA, supervision construction starts from the entity name or anchor. Benchmark evaluation questions and answers are used exclusively for evaluation, without entering supervision generation, filtering, or training. Section B gives the AERIS construction prompts.

Fine-tuning. All unlearning pipelines are implemented with LoRA on NVIDIA GeForce RTX 4090 GPU. We use LoRA rank r=8 (RWKU) and r=16 (TOFU), with α=16. Within each unlearning strategy, we fix the learning rate across forget-set sources and adjust epochs for dataset-size differences to approximately match the number of optimization steps. For RWKU, we follow ELUDe's tuning protocol with LoRA-stable learning rates. GA uses 8×10^{-6} for 4 epochs, NPO uses 5×10^{-5} for 4 epochs, and GA+GD / GA+KL use 5×10^{-6} for 3 epochs, while epochs for other forget-set sources are scaled to match steps. For TOFU, we fix the learning rate to 2×10^{-4} across all settings and tune epochs to keep optimization steps comparable. The TOFU benchmark supervision uses 4 (GA), 6 (NPO), 3 (GA+GD), and 6 (GA+KL) epochs, with corresponding epoch adjustments for DirectQA and AERIS.

A.2 DirectQA Construction Protocol

DirectQA constructs QA supervision directly from the target model, following Ma et al. (2025). Candidate pairs undergo five self-verification rounds and are accepted only when all rounds endorse the answer. The target QA count is set according to the scale of the reference supervision for the corresponding dataset, and candidate generation continues until that count is reached. DirectQA uses Wikipedia retain data under GA+GD and GA+KL. This repeated candidate-generation and verification loop builds an entity-specific QA collection; AERIS organizes elicited associations through recursive graph expansion and path-based synthesis. The self-verification prompt is provided in Figure 11 of Ma et al. (2025).

A.3 Method Hyperparameters

Association recovery. We build the local memory graph by iterative, target-conditioned elicitation and frequency-based filtering. We use N=10 elicitation queries per anchor, keep a candidate node if its elicitation-consistency score satisfies $s(\cdot) \geq \tau$, with τ=0.2 on RWKU and τ=0.3 on TOFU. We expand the graph up to K hops, with K=2 on RWKU and K=3 on TOFU.

Scoped Supervision Construction. We set the exploration coefficient to α=1, the maximum walk length to L=5, and the path-quality threshold to η=0.3. We run R=200 walks for RWKU to target the supervision scale of ELUDe, and R=20 walks for TOFU to match its recovered profile scale.

Figure 6 reports pairwise RWKU entity-overlap and frequency-similarity values.

Supervision-source pair	Entity overlap (Top-50 Jaccard)	Frequency similarity (cosine)
RWKU / ELUDe	0.25	0.27
RWKU / DirectQA	0.33	0.69
RWKU / AERIS	0.37	0.38
ELUDe / DirectQA	0.16	0.67
ELUDe / AERIS	0.27	0.86
DirectQA / AERIS	0.27	0.61

Figure 6: Average pairwise entity overlap and frequency-distribution similarity across RWKU. Each unordered pair is shown once; self-comparisons are omitted. The two measures distinguish overlap in the most frequent entities from agreement in their frequency profiles.

Finetune	Method	Forget Set ↓				Neighbor Set ↑			MIA Set		Utility Set ↑				
		FB	QA	AA	All	FB	QA	All	FM ↑	RM ↓	MMLU	BBH	TruthfulQA	TriviaQA	AlpacaEval
Before		0.6729	0.7290	0.6839	0.6953	0.6251	0.5819	0.6035	1.8448	1.9326	0.6864	0.4090	0.3775	0.3985	6.0196
GA	RWKU	0.6597	0.7404	0.6804	0.6935	0.6227	0.5692	**0.5959**	1.8645	1.9411	**0.6879**	0.3133	0.3800	**0.4077**	6.0383
	ELUDe	0.6694	0.6799	0.6625	0.6706	0.6113	**0.5793**	0.5953	1.8567	**1.9383**	0.6857	0.3148	0.3925	0.2670	**6.1704**
	DirectQA	0.6675	**0.6541**	**0.6296**	0.6504	**0.6377**	0.5534	0.5956	**2.0591**	2.0101	0.6784	0.4336	0.3775	0.3506	6.0679
	AERIS	**0.6236**	0.6724	0.6355	**0.6438**	0.6083	0.5590	0.5837	1.8663	1.9439	0.6842	**0.4383**	**0.3975**	0.2460	6.1535
NPO	RWKU	**0.6245**	0.7129	0.6561	0.6645	0.6300	**0.5688**	**0.5994**	**2.1448**	2.1175	0.6827	0.4012	0.3900	0.2253	6.2853
	ELUDe	0.6793	**0.5073**	0.4865	0.5577	0.5971	0.4836	0.5404	1.9096	1.9610	0.6842	0.3858	**0.4475**	0.1546	**6.4073**
	DirectQA	0.6497	0.6870	0.6826	0.6731	0.6183	0.5561	0.5872	1.9156	1.9630	**0.6849**	0.4213	0.3875	**0.3771**	6.0951
	AERIS	0.6283	0.5097	**0.4550**	**0.5310**	**0.6348**	0.5491	0.5920	1.8984	**1.9548**	0.6835	**0.4290**	0.4325	0.1642	6.3429
GA+GD	RWKU	0.6356	0.7296	0.6675	0.6776	0.5952	**0.5772**	**0.5862**	**1.8965**	1.9507	0.6871	0.3441	0.3750	0.4245	6.0307
	ELUDe	0.6356	**0.6163**	**0.6440**	**0.6386**	**0.6084**	0.5624	0.5854	1.8513	**1.9342**	**0.6893**	0.2238	**0.3975**	0.2859	**6.1453**
	DirectQA	0.6422	0.7201	0.6545	0.6723	0.5827	0.5702	0.5765	1.8635	1.9436	0.6820	0.3441	0.3825	**0.4515**	5.8489
	AERIS	0.6484	0.7042	0.6515	0.6680	0.5995	0.5646	0.5821	1.8592	1.9413	0.6857	**0.4306**	0.3825	0.4408	5.9923
GA+KL	RWKU	**0.6293**	0.7296	0.6684	0.6758	0.6131	**0.5746**	0.5938	**1.8951**	1.9506	0.6849	0.3997	0.3750	**0.4113**	6.0506
	ELUDe	0.6828	0.6847	0.6722	0.6799	0.6072	0.5710	0.5891	1.8559	1.9376	**0.6879**	0.4090	**0.3950**	0.2730	**6.1676**
	DirectQA	0.6608	**0.6529**	**0.6535**	**0.6557**	**0.6685**	0.5491	**0.6088**	1.8807	1.9429	0.6857	0.3426	0.3850	0.3752	6.0251
	AERIS	0.6330	0.7135	0.6694	0.6720	0.5808	0.5691	0.5749	1.8535	**1.9370**	0.6864	**0.4306**	0.3850	0.2969	6.1099

Table 11: Unlearning performance with different fine-tuning strategies and forget-set generation methods on Phi-3-Mini-4K-Instruct. Bold denotes the best result and underlined denotes the second best. Arrows indicate the direction of preference (↓ lower is better; ↑ higher is better).

A.4 Evaluation Metrics

A.4.1 RWKU Evaluation Metrics

RWKU (Jin et al., 2024) reports results in four groups, Forget Set (↓), Neighbor Set (↑), MIA Set, and Utility Set (↑).

Forget Set (↓). RWKU measures whether a model still reveals target knowledge using three probe types, (i) FB (fill-in-the-blank) probes that mask knowledge points in sentences from the target's Wikipedia page and ask the model to complete the blank, (ii) QA (question-answer) probes that directly query target-related facts, and (iii) AA (adversarial-attack) probes that attempt to elicit forgotten knowledge via prompt attacks (e.g., prefix injection, affirmative suffix, role playing, reverse query, synonym manipulation, background hints, in-context demonstrations, and cross-lingual queries). For FB/QA/AA, RWKU uses ROUGE-L recall between the model output and the ground-truth answer. Lower is better for unlearning efficacy. We additionally report All as the aggregate over the three forget subsets, consistent with RWKU's reporting format.

Neighbor Set (↑). To assess locality (i.e., not over-forgetting beyond the target), RWKU constructs neighbor probes that are closely related to the target but outside the intended forgetting scope. Similar to the forget set, it includes Neighbor FB and Neighbor QA, evaluated by ROUGE-L recall. Here higher is better, indicating less collateral damage on neighboring knowledge. We report All as the aggregate over the neighbor subsets.

MIA Set (FM ↑, RM ↓). RWKU further evaluates whether target knowledge still appears as pre-training members via membership inference attacks (MIAs). It defines a forget-member set (FM) containing target-related training fragments and a retain-member set (RM) with unrelated member fragments. RWKU provides four MIA scorers (LOSS, Zlib entropy, Min-K% Prob, and Min-K%++ Prob) and primarily reports LOSS-based scores. Higher scores indicate the text is less likely to be memorized. Therefore, effective unlearning should yield higher FM (target fragments look non-

member) while keeping lower RM (unrelated member fragments still look like members), matching the (FM ↑, RM ↓) convention.

Utility Set (↑). RWKU measures general capability preservation with five standard tasks with higher scores preferred, namely MMLU (Gen) using 5-shot accuracy via answer perplexity, BBH (Rea) using chain-of-thought prompting with 3-shot examples and exact-match scoring, TruthfulQA (Tru) using MC1 6-shot accuracy, TriviaQA (Fac) using 6-shot F1, and AlpacaEval (Flu) using the weighted average of bi-/tri-gram entropies as a fluency proxy.

A.4.2 TOFU Evaluation Metrics

TOFU (Maini et al., 2024) evaluates unlearning with four splits, Forget Set (target to forget), and three non-forget splits for utility preservation—Retain Set, Real Authors, and World Facts—forming a relevance gradient from near-domain to far-domain (Maini et al., 2024).

Forget metrics. On the Forget Set, we report (i) Prob↓, the length-normalized conditional likelihood of the ground-truth answer under the model, computed as $P(a \mid q)^{1/|a|}$ (lower is better when forgetting) (Maini et al., 2024); (ii) ROUGE↓, the ROUGE-L recall between the greedy-decoded answer and the ground-truth answer (lower indicates less recoverable target content) (Maini et al., 2024); (iii) TruthRatio↑, a likelihood-ratio style score that contrasts a paraphrased correct answer $\tilde{a}$ against a set of GPT-generated perturbed (factually incorrect) answers $\mathcal{A}_{\text{pert}}$, defined as $\log R_{\text{truth}} = \frac{1}{|\mathcal{A}_{\text{pert}}|} \sum_{\hat{a} \in \mathcal{A}_{\text{pert}}} \frac{1}{|\hat{a}|} \log P(\hat{a} \mid q) - \frac{1}{|\tilde{a}|} \log P(\tilde{a} \mid q)$, where $\tilde{a}$ and $\mathcal{A}_{\text{pert}}$ are constructed to control for phrasing effects (Maini et al., 2024), and (iv) Forget Q.↑, the forget quality score computed via a two-sample KS-test on the TruthRatio distributions of the unlearned model versus a retain-only model, using the resulting p-value as the forgetting score (higher values indicate less evidence of a distributional difference under this test) (Maini et al., 2024). Consistent with the TOFU convention, Forget Set uses "lower Prob/ROUGE, higher TruthRatio" as the desired direction (Maini et al., 2024).

Utility metrics. For utility preservation, TOFU rescales metrics so that higher is better on non-forget splits (e.g., converting TruthRatio by $\max(0, 1 - R_{\text{truth}})$ on non-forget sets) and computes Model Utility↑ as the harmonic mean of three metrics (Prob/ROUGE/TruthRatio) across the three non-forget splits (Retain, Real Authors, World Facts), yielding nine values in total. The harmonic mean penalizes any single metric collapse (Maini et al., 2024). In addition, we report per-split summary scores in the same order as the table header, RS Score↑ (Retain Set), RAS Score↑ (Real Authors Set), and WFS Score↑ (World Facts Set). Each score is computed on its corresponding split as the harmonic mean of Prob., ROUGE, and an accuracy-oriented term (Acc.), so that higher indicates better overall utility on that split.

Complementary views of forgetting. Memorization Accuracy and Extraction Likelihood directly test whether target sequences remain recoverable (Jang et al., 2023). RWKU FB/QA/AA probes extend this view to direct and adversarial queries, RWKU MIA scores measure membership-related privacy risk, and TOFU Prob/ROUGE/TruthRatio/Forget Quality compare target recoverability against a retain-only reference. Together, these benchmark-native metrics cover extraction, membership, and behavioral forgetting.

A.5 Basic Fine-tuning Strategies

AERIS supplies forget and neighbor supervision to the following unlearning pipelines.

Gradient Ascent (GA) inverts the standard training objective for forget set S_f by ascending the gradient of the loss function as

$$\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{L}_{\text{LM}}(S_f)$$

Here, θ represents the model's parameters, η is the unlearning rate (a hyperparameter that controls the step size), and $\nabla_\theta \mathcal{L}_{\text{LM}}(S_f)$ is the gradient of the language model loss calculated on the forget set S_f.

Baseline	Generation Method	Forget				Utility			
		Prob.↓	ROUGE↓	TruthRatio↑	Forget Q.↑	RS Score↑	RAS Score↑	WFS Score↑	Model Utility↑
Before		0.9951	0.9493	0.5306	0.0013	0.7133	0.1667	0.0508	0.6277
GA	TOFU	**0.0353**	**0.0714**	**0.5903**	**0.1650**	0.1906	**0.1922**	**0.0725**	0.3309
	DirectQA	0.4260	0.4521	0.5029	0.0286	**0.5378**	0.0830	0.0212	**0.5783**
	AERIS	0.1492	0.4500	0.5285	0.0541	0.4449	0.0046	0.0024	0.5155
NPO	TOFU	**0.0359**	**0.0338**	**0.6272**	**0.1650**	0.1638	**0.2402**	**0.0948**	0.2951
	DirectQA	0.4393	0.4933	0.5637	0.0541	**0.5204**	0.0817	0.0220	**0.5743**
	AERIS	0.0732	0.2487	0.5670	0.0971	0.2613	0.0384	0.0046	0.3949

Table 12: Full GA and NPO results on TOFU-finetuned LLaMA-2-7B-Chat.

Baseline	Generation Method	Forget				Utility			
		Prob.↓	ROUGE↓	TruthRatio↑	Forget Q.↑	RS Score↑	RAS Score↑	WFS Score↑	Model Utility↑
Before		0.9951	0.9493	0.5306	0.0013	0.7133	0.1667	0.0508	0.6277
GA+GD	TOFU	**0.1031**	**0.1631**	**0.7365**	**0.2657**	0.1834	**0.5251**	**0.3904**	0.3497
	DirectQA	0.9843	0.9198	0.5395	0.0068	**0.7015**	0.1852	0.0616	**0.6333**
	AERIS	0.1608	0.4416	0.5283	0.1650	0.4748	0.0143	0.0048	0.5436
GA+KL	TOFU	0.4709	0.4409	**0.5644**	0.0541	**0.5923**	**0.1617**	**0.0516**	**0.5943**
	DirectQA	0.2916	0.4092	0.4950	0.0541	0.5069	0.0920	0.0272	0.5688
	AERIS	**0.1580**	**0.3359**	0.4716	**0.0971**	0.4767	0.0057	0.0028	0.5362

Table 13: Unlearning performance(GA+GD, GA+KL) on TOFU-finetuned LLaMA-2-7B-Chat. (↓ lower is better; ↑ higher is better)

Negative Preference Optimization (NPO) trains the model to treat the generation of forgotten content as a rejected behavior, contrasting it with the behavior of a reference model according to

$$\mathcal{L}_{\text{NPO}} = -\mathbb{E}_{x\sim S_f}\left[\log\sigma\left(\beta\log\frac{\pi_{\theta_{\text{ref}}}(y|x)}{\pi_\theta(y|x)}\right)\right]$$

Here, π_θ is the policy of the model being trained, $\pi_{\theta_{\text{ref}}}$ is the policy of a frozen reference model (usually the model state before unlearning), β is a temperature parameter controlling the strength of the preference, and σ is the sigmoid function.

We also consider two preservation objectives.

Gradient Descent (GD) applies the standard training objective on a curated retain set S_r, to encourage preservation of non-target behavior as

$$\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_{\text{LM}}(S_r)$$

Here, S_r is the retain set and θ represents the model's parameters.

Kullback–Leibler Divergence (KL) acts as a regularization constraint to maintain model utility. It penalizes the deviation of the updated model's policy π_θ from that of a static reference model $\pi_{\theta_{\text{ref}}}$ on the retain set as

$$\mathcal{L}_{\text{KL}} = \mathbb{E}_{x\sim S_r}\left[D_{\text{KL}}(\pi_{\theta_{\text{ref}}}(y|x)||\pi_\theta(y|x))\right]$$

Here, D_{KL} denotes the Kullback–Leibler divergence, which measures the difference between the probability distributions of the reference policy $\pi_{\theta_{\text{ref}}}$ and the current model policy π_θ over the retain set S_r.

Each forgetting objective can be paired with a utility-preservation objective. This design tests recovered supervision across downstream objectives.

B Prompt Template

B.1 Prompt for internal memory mining

Prompt template for E_t's 0-hop neighbor information mining.

Baseline	Generation Method	Forget				Utility			
		Prob.↓	ROUGE↓	TruthRatio↑	Forget Q.↑	RS Score↑	RAS Score↑	WFS Score↑	Model Utility↑
Before		0.7742	0.6469	**0.5077**	0.0068	0.5664	**0.2048**	0.1236	0.5735
GA	TOFU	0.2995	**0.3632**	**0.6180**	**0.2657**	0.4173	**0.5867**	**0.5741**	0.5222
	DirectQA	0.5684	0.4337	0.5530	0.0143	**0.5033**	0.2418	0.1948	**0.5586**
	AERIS	**0.2799**	0.5457	0.4635	0.0286	0.4531	0.0059	0.0028	0.4948
NPO	TOFU	**0.2197**	**0.3577**	**0.6328**	**0.0971**	0.3290	0.1771	0.0832	0.4371
	DirectQA	0.7731	0.6452	0.5094	0.0068	**0.5649**	**0.2088**	**0.1271**	**0.5733**
	AERIS	0.4493	0.6323	0.5033	0.0286	0.5365	0.0239	0.0097	0.5441
GA+GD	TOFU	**0.2912**	**0.3154**	**0.7063**	**0.2657**	0.3342	**0.5469**	**0.4962**	0.4592
	DirectQA	0.5588	0.4149	0.5750	0.0068	0.4463	0.4221	0.3572	**0.5230**
	AERIS	0.3165	0.5495	0.5030	0.1650	**0.4894**	0.0096	0.0043	0.5096
GA+KL	TOFU	**0.1674**	**0.2680**	0.4734	**0.4046**	0.4772	0.1876	0.1108	0.5453
	DirectQA	0.6929	0.5873	**0.5138**	0.0068	**0.5576**	**0.2068**	**0.1357**	**0.5749**
	AERIS	0.5174	0.5727	0.4741	0.0541	0.5467	0.0227	0.0121	0.5449

Table 14: Unlearning performance with different forget set generation methods on TOFU using a TOFU-finetuned LLaMA-3.2-1B-Instruct model. (↓ lower is better; ↑ higher is better)

Attributes	**Freq**	**Gold**	**DirectQA**			**AERIS**		
			Pred	**F1**	**Covered**	**Pred**	**F1**	**Covered**
BirthPlace	2	kuwait city, kuwait	kuwait city, kuwait;	1.00	✓	kuwait city;	0.80	✓
Gender	1	male		0			0	
BirthYear	1	1956		0		1931; 1956;	0.67	✓
Genre	3	french literature	Arabic culture and Middle Eastern landscapes;	0	✓	french literature; literary fiction;	0.67	✓
Awards	1	prix goncourt		0		prix goncourt;	1.00	✓
Par_Job	4	florist; game developer;	florist; game developer; accountant;	0.80	✓		0	
Books	3	promise by the seine; le petit sultan;		0		promise by the seine; le petit sultan; the parisian promise;	0.80	✓
Books	1	river of jasmine; the nightingale of paris;		0		the oasis of paris; mirror of paris; the nightingales song; the nightingale of paris;	0.33	✓

Table 15: Qualitative comparison of recovered attributes against TOFU ground-truth profiles.

```
Recall information from your own internal knowledge.
Target: {target}
Write 5-10 atomic statements about {target}.
```

Prompt template for h-hop neighbor information mining, given the $h-1$-hop neighbor entity and the unlearning target.

```
Recall information from your own internal knowledge.
Target: {target}
Neighbor: {neighbor}
Write 5-10 SHORT atomic statements specifically about how
{neighbor} relates to {target}.
```

B.2 Prompt for corpus construction

Prompt template for event statement construction, given the two neighbor entities from sampled path and the unlearning target.

```
Please act as an information assistant to help users learn
about pertinent details regarding the target.
Given an anchor target and two key events related to it, use
known knowledge about the target to provide ONE concise
factual statement.
The statement should highlight details about the {target} that
users may find important.
Do NOT invent fictional or hypothetical scenarios.  If you are
not confident the connection is real, output UNKNOWN.
The statement should explicitly include BOTH event names (do
not use pronouns), and keep it to ONE sentence.
Anchor Target:  {target}
Event 1:  {event_1}
Event 2:  {event_2}
```

Prompt template for QA-style forget set construction, given the event statement, context obj and the unlearning target.

```
Please act as an information assistant to help users learn
about pertinent details.
Given a factual statement about the target, rewrite it into
ONE QA pair.
The question should highlight an important detail from the
statement about the Central Context.
The question MUST NOT include the target entity's name or
directly reveal the answer.
Target Entity:  {target}
Central Context (Obj):  {obj}
Statement:  {event}
```

C More Results

C.1 Unlearning Results in Large Model

Table 12 reports the complete GA and NPO metrics summarized by the compact main-paper table. Table 13 further details GA+GD and GA+KL on LLaMA-2-7B-Chat. Under GA+GD, AERIS raises Forget Q. from DirectQA's 0.0068 to 0.1650 while maintaining model utility of 0.5436. Under GA+KL, AERIS achieves the lowest Prob and ROUGE scores and the highest Forget Q., extending the forgetting gains to KL-regularized updates.

C.2 Unlearning Results in Smaller Model

C.2.1 RWKU Evaluation on Phi-3-Mini-4K-Instruct

Table 11 shows results on Phi-3-Mini-4K-Instruct. AERIS achieves the lowest aggregate Forget Set score under GA and NPO. Under GA+GD, it also maintains BBH at 0.4306 while ELUDe reaches 0.2238, combining target forgetting with stronger reasoning preservation.

C.2.2 TOFU Evaluation on LLaMA-3.2-1B-Instruct

Table 14 presents the unlearning results on LLaMA-3.2-1B-Instruct. Across GA, NPO, GA+GD, and GA+KL, AERIS raises Forget Q. from DirectQA's scores of 0.0143, 0.0068, 0.0068, and 0.0068 to 0.0286, 0.0286, 0.1650, and 0.0541, respectively. Under GA, it also obtains the lowest target-answer probability (0.2799). The four-objective pattern shows that mined internal memory supplies an effective forgetting signal in the smaller architecture.

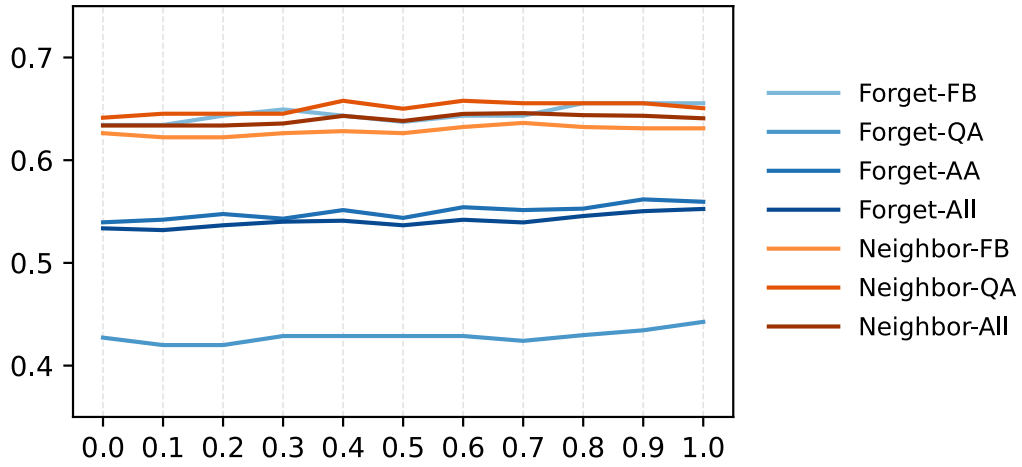


Figure 7: Impact of the proportion of correct statements in the forget set on unlearning performance.

D Additional Analysis

D.1 TOFU Recovery Details

The main paper reports the compact attribute-coverage comparison. This subsection gives the corresponding answer-level details for "Basil Mahfouz Al-Kuwaiti" in Table 15.

D.2 Correctness Mixture

We use GPT-4 to label the factual correctness of complete, naturally generated QA pairs in the RWKU forget set. Correct and incorrect pairs form two sampling pools. We draw fixed-size QA mixtures with correct-pair fractions from 0% to 100% and apply GA to each mixture, keeping the remaining training settings fixed. Figure 7 shows comparatively small changes across the mixing ratios. The forget objective is applied to the target answers.

D.3 Variation Across Models

For Taylor Swift in RWKU, Figure 8 compares associations elicited from LLaMA-2-7B-Chat, LLaMA-3-8B-Instruct, and GPT-4o. The displayed lists share biographical anchors such as December 13, 1989 and American, alongside model-specific entries. Taylor Alison Swift appears in the LLaMA-3-8B-Instruct list but not in the other two displayed lists. The main-paper ranking view also shows works such as Fearless and associated people such as Scott Borchetta among the model-specific entries. These observed differences motivate constructing supervision for the model being updated. The heatmap reports association scores from repeated QA elicitation for this target, with ordering and shading interpreted within each model. These raw scores use a separate scale from the normalized response frequency $s(v)$ in Eq. 3.

D.4 Knowledge Coverage

On RWKU with `LLaMA-2-7B-Chat`, we vary the coverage setting during forget-set construction while keeping the QA-set size approximately constant. For target t, coverage is the number of distinct graph entities represented in the generated QA divided by the total number of graph entities. We compute this ratio separately for each target and average across targets. Figure 9 reports performance at mean coverage values 0.45, 0.57, 0.68, 0.82, 0.86, and 0.91. Forget-All varies across these settings, while Neighbor-All remains within a narrower range.



Figure 8: One-hop associations from repeated QA elicitation for the same target. Raw scores describe within-model association rankings.

D.5 Memory Graph Stability Analysis

We compare graphs recovered from the same target model under changes to the elicitation and filtering settings.

Specifically, with the underlying model fixed, we perturb four factors during graph construction, namely elicitation prompts, random seeds, generation temperature, and memory filtering threshold. The reference setting uses temperature 0.6 and τ=0.2. The temperature and threshold perturbations use 0.9 and τ=0.6, respectively. For each perturbed setting, we compare the resulting and original graphs using local and global similarity metrics.

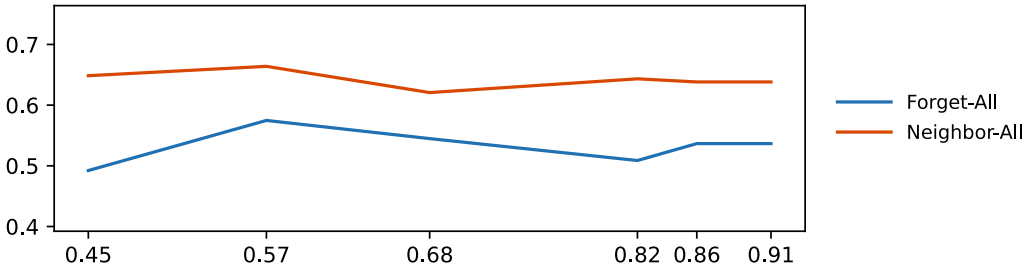


Figure 9: RWKU forgetting and neighbor preservation across mean graph-coverage settings at approximately constant QA-set size. The vertical axis reports ROUGE-L recall.

Setting	Nbr.	Rank	Head	NMI	2-hop
Prompt	0.75±0.08	0.85±0.09	0.49±0.04	0.48±0.06	0.48±0.04
Seed	0.82±0.04	0.82±0.19	0.50±0.04	0.48±0.06	0.49±0.05
Temp. (0.9)	0.66±0.11	0.77±0.17	0.46±0.05	0.43±0.05	0.42±0.06
Threshold (τ=0.6)	0.62±0.11	0.84±0.12	0.23±0.07	0.59±0.12	0.26±0.07

Table 16: Graph stability under construction perturbations, reported as mean ± standard deviation across entities. Nbr., Rank, Head, NMI, and 2-hop denote Root-Nbr Jacc., Root-Rank, Head-Node Jacc., Comm. NMI, and 2-hop Ego Jacc., respectively.

We evaluate graph stability using five complementary metrics, (i) Root-Nbr Jacc., which measures the Jaccard similarity of one-hop neighbors around the target entity, (ii) Root-Rank ρ, which measures ranking consistency among shared neighbors, (iii) Head-Node Jacc., which measures the overlap of high-importance nodes in the graph, (iv) Comm. NMI, which measures the consistency of community assignments, and (v) 2-hop Ego Jacc., which measures the overlap of the target-centered two-hop subgraph.

Table 16 separates rank agreement from set membership. Shared-neighbor rank correlations range from 0.77 to 0.85. For the stricter threshold, root-neighbor, head-node, and two-hop Jaccard scores are 0.62, 0.23, and 0.26. Thus stable rankings among shared nodes coexist with changes in the nodes retained. The table reports means and standard deviations of the graph-comparison metrics across target entities.

D.6 External Knowledge Alignment Analysis

We assess external recognizability by aligning recovered entities with a reference knowledge graph.

FaithUn constructs knowledge-graph-derived probes for paraphrased, multi-hop, and same-answer knowledge (Yang et al., 2025). We extend this graph-based evaluation perspective to the broader neighborhood recovered from the target model. Specifically, for each RWKU entity, we use all incoming and outgoing one-hop Wikidata neighbors as the reference.

We compare three sources, (i) ELUDe-generated forget sets, (ii) entities contained in the memory graph recovered by AERIS (AERIS-Graph), and (iii) entities extracted from the synthesized forget data (AERIS-Response).

Entity Coverage measures the fraction of model-side entities matched to Wikidata. Wiki Coverage measures the fraction of Wikidata neighbors covered by the generated entities. Results are summarized as mean ± standard deviation across target entities.

Table 17 reports Entity Coverage of 0.37 for AERIS-Graph and 0.20 for ELUDe: a larger fraction of the graph entities matches the reference neighborhood. AERIS-Response reaches 0.45 Wiki Coverage, compared with 0.32 for the raw graph, so its synthesized data cover a larger fraction of the Wikidata neighbors. Entity Coverage and Wiki Coverage use the generated set and the reference set as their respective denominators; the table reports both for each source.

Method	Entity Coverage ↑	Wiki Coverage ↑
ELUDe	0.20±0.06	0.52±0.13
AERIS-Graph	0.37±0.08	0.32±0.12
AERIS-Response	0.19±0.05	0.45±0.13

Table 17: External recognizability against Wikidata one-hop neighborhoods, summarized as mean ± standard deviation across entities.

D.7 Low-Pageview Entities and Input Conditions

We use Wikipedia pageviews in the range 20–50 to select less prominent entities for an exploratory analysis of the input conditions for association recovery. Elicitation produced empty graphs when the model declined to answer or did not identify the entity. For other targets, one-hop association frequencies followed a long-tail distribution, with few associations recurring across responses.

D.8 Artifact Identifiers

The TOFU checkpoints used in our experiments are:

- LLaMA-2-7B-Chat: `https://huggingface.co/locuslab/tofu_ft_llama2-7b`.
- LLaMA-3.2-1B-Instruct: `https://huggingface.co/open-unlearning/tofu_Llama-3.2-1B-Instruct_full`.