MAKING LLMS TRULY FORGET: DEEP UNLEARNING BY SEARCHING, SELECTING, AND SEVERING KNOWLEDGE PATHS

Anonymous authors
Paper under double-blind review

ABSTRACT

While an unlearned language model may no longer recall a fact directly, the fact often remains recoverable through multi-hop reasoning over related knowledge. Most existing unlearning techniques overlook this vulnerability, targeting facts in isolation while leaving their supporting knowledge intact. To achieve true forgetting, we propose a general deep unlearning framework compatible with existing unlearning algorithms. Our approach adaptively explores both explicit responses and latent internal representations to discover valid reasoning paths, compiles them into a confidence-aware supporting subgraph, and we apply a graph minimum cut to sever all recovery paths while preserving unrelated knowledge. To rigorously evaluate deep unlearning, we introduce a model-specific pipeline that extracts and completes knowledge graphs from raw text, filtering them by calibrated model confidence to reflect what the model genuinely retains. Comprehensive experiments demonstrate that selectively unlearning supporting knowledge yields substantially deeper forgetting than superficial methods while preserving model utility, highlighting that genuine unlearning requires breaking the relational structures that enable factual reconstruction.

1 INTRODUCTION

Large language models (LLMs) acquire broad knowledge and capabilities from large-scale text corpora. However, these corpora may also contain copyrighted material or private information that models should not retain (Henderson et al., 2023; Min et al., 2024). Removing the influence of such information after training is challenging, as retraining the model from scratch without the target data is often prohibitively expensive (Eldan & Russinovich, 2023; Jang et al., 2023). Machine unlearning addresses this challenge by efficiently modifying a trained model to remove the influence of designated data while preserving its behavior on unrelated data (Cao & Yang, 2015; Sekhari et al., 2021; Wei et al., 2024).

Despite recent progress in factual unlearning, most existing methods still treat knowledge as a collection of isolated targets (Yao et al., 2024b; Maini et al., 2024; Shi et al., 2025; Jin et al., 2024; Ma et al., 2025). However, factual knowledge in LLMs is highly interconnected, and removing direct access to a fact does not necessarily make it unrecoverable: related knowledge may still support its reconstruction (Wei et al., 2025; Wang et al., 2026a). For instance, as illustrated in Figure 1, even if the model no longer directly recalls that "Ron and Ginny Weasley" are siblings, it may still retain that both are children of "Arthur and Molly Weasley", from which the sibling relation can be inferred. Thus, effective unlearning requires more than removing the target fact itself. It must also account for the supporting knowledge that keeps the target inferable. We refer to this stronger objective as deep unlearning (Wu et al., 2024; Wei et al., 2025).

To address this limitation, we propose Deep Unlearning via Minimum Inferential Cuts (DUMIC), a general framework compatible with existing unlearning algorithms. At a high level, given a target fact, our framework dynamically maps out both explicit dependencies and implicit associations retained by the model to identify the supporting knowledge that enables reconstruction. Specifically, DUMIC operates in three stages: search, select, and sever. During search, to capture inference paths beyond what surface text reveals, we integrate explicit model responses with latent entities decoded

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.

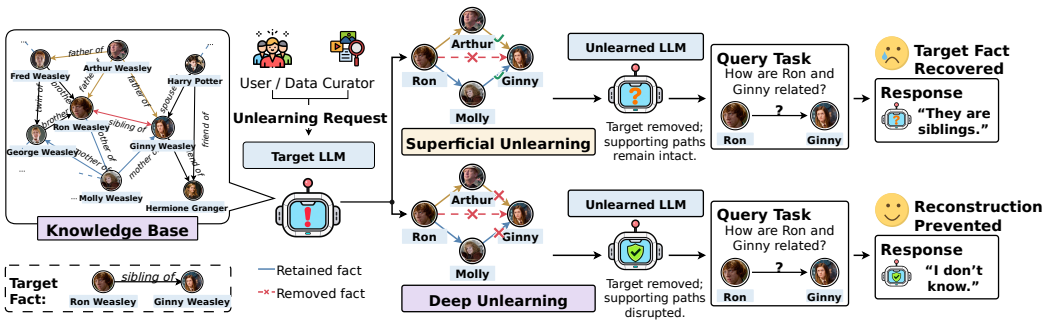

Figure 1: Illustration of deep unlearning. A target fact may remain recoverable through reasoning over related knowledge even after unlearning. Deep unlearning aims to interrupt the supporting knowledge structures that enable such recovery.

from intermediate-layer representations through a decoding procedure that maps these representations to potentially relevant entities that are excluded from the model's response. These additional entities expand the search frontier and uncover supporting paths that would otherwise remain unexplored. Using only explicitly mentioned entities may therefore miss paths through which the target fact can be reconstructed. During the select stage, we verify whether these candidate paths genuinely substantiate the target fact, filtering out invalid associations and calibrating model confidence to isolate the active supporting subgraph. Finally, in the severing stage, we formulate supporting-fact selection as a confidence-weighted graph minimum cut over this subgraph. This identifies the minimal-cost subset of facts necessary to break all verified recovery paths, achieving thorough unlearning while strictly preserving model utility.

To rigorously evaluate deep unlearning, we introduce a model-specific evaluation pipeline that assesses whether target knowledge remains recoverable from the facts retained by each model. Grounded in a comprehensive reference graph extracted and completed from raw text, our pipeline filters relational facts through calibrated confidence to mirror each model's actual knowledge state. Extensive experiments reveal three key findings. First, surface model responses conceal critical reasoning routes: latent entities decoded from intermediate representations substantially expand path coverage, exposing hidden avenues for factual reconstruction. Second, superficial unlearning offers only an illusion of forgetting, as leaving supporting knowledge intact allows target facts to remain reliably inferable via multi-hop reasoning. Third, DUMIC's confidence-aware minimum cut surgically severs these inferential paths, achieving deep forgetting while mitigating the catastrophic utility collapse incurred by unconstrained knowledge removal.

2 Related Work

Superficial Unlearning. Most LLM unlearning methods define a forget set and update the trained model to rely less on that data (Eldan & Russinovich, 2023; Kurmanji et al., 2023; Liu et al., 2024; Jia et al., 2024; Niu et al., 2025; Wang et al., 2026b). Gradient Ascent (GA) raises the loss on the forget set, which lowers the probability of the target responses (Jang et al., 2023; Yao et al., 2024a). However, unconstrained gradient ascent can severely damage knowledge outside the forget set (Kurmanji et al., 2023; Maini et al., 2024; Jia et al., 2024). Negative Preference Optimization (NPO) is more stable, because its bounded objective only lowers the likelihood of forget data relative to the original model (Zhang et al., 2024). Existing benchmarks test whether the target content is still accessible after unlearning (Eldan & Russinovich, 2023; Maini et al., 2024; Shi et al., 2025). TOFU asks biographical questions about fictitious authors (Maini et al., 2024), while MUSE tests memorized text under several forms of access (Shi et al., 2025). Both also measure retained utility, but their forgetting criterion only evaluates the target itself. They do not ask whether related knowledge can rebuild the target, which makes the evaluation superficial (Wei et al., 2025; Wang et al., 2026a).

Deep Unlearning. Deep unlearning requires a target fact to become uninferable from a model's retained knowledge, moving beyond the superficial suppression of direct recall (Wu et al., 2024; Wei et al., 2025). Prior studies formulate this objective using rule-based deduction on fixed knowledge

bases (Wu et al., 2024) or subgraphs guided by external graphs (Wei et al., 2025), demonstrating that erased facts often persist through logical associations and paraphrases (Wang et al., 2026a; Wu et al., 2025). A closely related line of research investigates relational knowledge via *multi-hop knowledge editing* (Zhong et al., 2023; Shi et al., 2024; Cohen et al., 2024; Cheng et al., 2025). However, the two paradigms operate in fundamentally opposite directions. Multi-hop editing examines a forward ("cause-to-effect") propagation: it updates an upstream premise and tracks whether its downstream logical consequences are updated accordingly. In contrast, deep unlearning tackles a reverse ("effect-to-cause") challenge: given a target fact slated for removal, it must trace backward to determine *which* supporting premises must be selectively severed to prevent the target from being reconstructed.

3 The Formulation of Knowledge Unlearning

Confidence-Aware Knowledge Modeling in LLMs. We represent the relational knowledge of a pretrained LLM $\mathcal{M}_{\text{pre}}$ as a confidence-aware graph $\mathcal{G}_{\mathcal{M}_{\text{pre}}} = (\mathcal{E}, \mathcal{R}, \mathcal{T}_{\mathcal{U}})$, where $\mathcal{E}$ and $\mathcal{R}$ denote the entity and relation sets, respectively. Each fact $t = (s, r, o, u) \in \mathcal{T}_{\mathcal{U}}$ consists of a subject, a relation, an object, and a calibrated confidence u, which reflects the model's confidence in the corresponding triple. Given a target triple $e = (s^*, r^*, o^*)$, the model may retain e directly or reconstruct it from related facts. A supporting path $P_i = (t_{i,1}, \ldots, t_{i,\ell_i})$ consists of facts forming a connected reasoning chain that supports e. We denote the set of supporting paths retained by the model for target e as $\mathcal{P}_{\mathcal{M}}(e)$. For example, $(\text{Ron}, \textsc{SiblingOf}, \text{Ginny})$ may be inferred from $(\text{Arthur}, \textsc{FatherOf}, \text{Ron}, u_1)$ and $(\text{Arthur}, \textsc{FatherOf}, \text{Ginny}, u_2)$.

LLM Unlearning. Let $\mathcal{M}_{\mathcal{A}}$ be the model obtained by applying an unlearning method $\mathcal{A}$ to $\mathcal{M}_{\text{pre}}$, and let $u_{\mathcal{A}}(t)$ denote its calibrated confidence in a fact t. For a target fact $e = (s^*, r^*, o^*)$, let $\mathcal{P}_{\mathcal{M}_{\text{pre}}}(e)$ be its verified supporting paths in the pretrained model, excluding e itself. Because reconstructing e through a path P requires every constituent fact, we define the path's confidence after unlearning as $C_{\mathcal{A}}(P) = \min_{t \in P} u_{\mathcal{A}}(t)$. Let $\mathcal{Q}_e \subseteq \mathcal{P}_{\mathcal{M}_{\text{pre}}}(e)$ denote the supporting paths considered in the unlearning objective. Given a threshold γ, $\mathcal{A}$ achieves deep unlearning of e with respect to $\mathcal{Q}_e$ if $u_{\mathcal{A}}(e) \leq \gamma$ and $C_{\mathcal{A}}(P) \leq \gamma$ for every $P \in \mathcal{Q}_e$. The complete objective takes $\mathcal{Q}_e = \mathcal{P}_{\mathcal{M}_{\text{pre}}}(e)$, requiring every verified supporting path to be weakened. Superficial unlearning is the special case $\mathcal{Q}_e = \varnothing$, in which the same criterion reduces to $u_{\mathcal{A}}(e) \leq \gamma$. At the subgraph level, a set of supporting facts C cuts all paths under consideration if $C \cap P \neq \varnothing$ for every $P \in \mathcal{Q}_e$.

4 Methodology

4.1 Overview of the Deep Unlearning Framework

Building on the definition of deep unlearning in Section 3, we seek to remove a target fact e and interrupt its reconstruction from knowledge retained by the pretrained model $\mathcal{M}_{\text{pre}}$. As illustrated in Figure 2, DUMIC consists of three stages: (1) Adaptive Tree Search combines entities explicitly generated in model responses with entities mapped from representations at the model's intermediate layers. These mapped entities are crucial for discovering supporting paths that response-only search misses, as shown in Section 6.2;(2) Because discovered paths may be invalid, Supporting Path Pruning verifies and filters the candidates to form a supporting subgraph $\mathcal{G}_e$. (3) To interrupt the supporting paths while preserving as much other knowledge as possible, Minimum Cut Selection identifies a compact set of facts that intersects every path in $\mathcal{G}_e$ at minimum confidence-based cutting cost.

4.2 Discovering Supporting Paths via Adaptive Tree Search

Given a target triple $e = (s^*, r^*, o^*)$, our goal is to discover candidate supporting paths encoded in the pretrained model $\mathcal{M}_{\text{pre}}$ by exploring relational connections from s^* toward o^* within a bounded path length. We organize exploration as a dynamically constructed search tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ contains search nodes and $\mathcal{E}$ records the expansion edges between them. Each node $n = (v, \pi)$ stores the current entity v and an ordered sequence of relational triples π connecting s^* to v, with the root defined as $n_0 = (s^*, \varnothing)$. The same entity reached through different paths corresponds to distinct search nodes. Let $\mathcal{F}_d$ denote the frontier of nodes at depth d awaiting expansion, each with

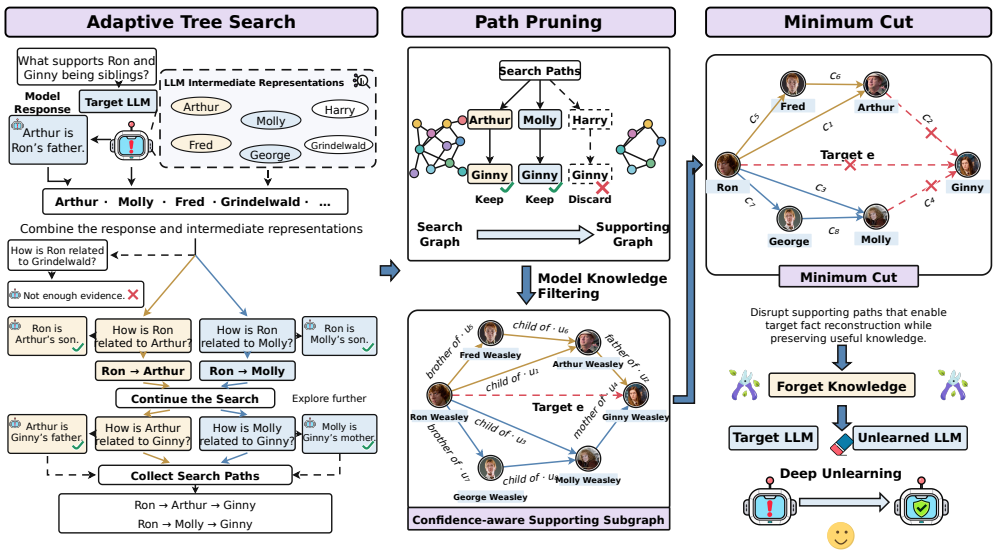

Figure 2: Overview of the proposed deep unlearning framework. Adaptive tree search discovers candidate supporting paths, path pruning constructs a confidence-aware supporting subgraph, and minimum-cut selection disrupts these paths while preserving model utility.

a path of d edges. Starting from $\mathcal{F}_0 = \{n_0\}$, we perform breadth-first search by applying two stages to each node in the current frontier: candidate entity discovery and relational edge verification.

Candidate Entity Discovery. For each node $n = (v, \pi) \in \mathcal{F}_d$, we query $\mathcal{M}_{\text{pre}}$ for facts related to v and extract the entities explicitly mentioned in its response. Since exhaustively identifying all related knowledge encoded in the model is infeasible, we seek to broaden search coverage by supplementing explicit responses with entity candidates decoded from intermediate representations. Specifically, we use the Jacobian lens (J-Lens) (Gurnee et al., 2026) to identify latent entities encoded in the model's intermediate representations but absent from its explicit response, providing additional candidates for extending π. However, decoding intermediate representations yields numerous candidate tokens, including noisy or uninterpretable fragments. To efficiently select relevant candidates, we first apply a probability threshold to reduce this noise, then rank the remaining tokens by their readout probabilities to prioritize candidates likely to be associated with the current entity v. We further resolve the selected tokens into entity names for subsequent relational edge verification.

Relational Edge Verification. An entity's appearance in the response or intermediate representations does not establish its relational connection to v. For each candidate entity w, we therefore query $\mathcal{M}_{\text{pre}}$ to determine whether a connecting relation exists and, if so, identify that relation. When a triple (v, r, w) is returned, we append it to π to obtain an extended path π' and add a new search node $n' = (w, \pi')$ connected to n. Candidates for which no relation is obtained do not extend the search tree. To reduce search overhead, we simplify expansion at the final permitted hop. Since the next entity must be o^* to complete a path within the remaining depth budget, we skip candidate entity discovery and directly verify the connection between each frontier entity v and o^*. If a triple (v, r, o^*) is returned, we append it to the current path. This avoids further entity generation, intermediate-representation decoding, and relation queries for alternative candidates.

Whenever an extended path π' reaches o^* and satisfies the minimum path length, we record it as a completed candidate path. Otherwise, if the path remains eligible for expansion, we add the corresponding node n' to the next frontier $\mathcal{F}_{d+1}$. Once all nodes in $\mathcal{F}_d$ have been processed, we repeat the expansion procedure for the next frontier. The search terminates when the maximum depth is reached or the frontier becomes empty. All completed candidate paths are then passed to the pruning stage to assess whether their constituent relations jointly support the target fact and to remove invalid or redundant paths.

4.3 Pruning Supporting Paths for Subgraph Refinement

Connectivity between s^* and o^* does not ensure that the relations along a path support the target fact e. We therefore use LLM to assess whether the constituent triples jointly support the inference

of e, restricting its judgment to the supplied facts. We discard unsupported paths and construct a supporting subgraph by taking the union of triples across the retained paths. We then determine which facts in this subgraph are retained by $\mathcal{M}_{\text{pre}}$. To obtain reliable confidence estimates for this filtering step, we calibrate the model's output probabilities to better align its confidence with empirical accuracy. Each unique triple is converted into a multiple choice question with options A, B, C, D, and Unknown. A fact passes the filter only when the model selects the correct answer and the associated calibrated confidence satisfies the criterion.

4.4 Interrupting Supporting Paths via Confidence-Aware Minimum Cut

Effective deep unlearning requires not only preventing the recovery of target knowledge, but also preserving as much unrelated knowledge as possible. Directly unlearning every fact in $\mathcal{G}_e$ may reliably interrupt its supporting paths, but can cause unnecessary utility loss. This motivates selecting a compact subset of supporting facts that interrupts all recovery paths while minimizing the expected knowledge loss. We formulate supporting-fact selection as a confidence-aware minimum cut on $\mathcal{G}_e$, which excludes the target fact e. Removing a set of facts C interrupts a path if at least one fact on that path belongs to C; C constitutes a cut if its removal interrupts every path from s^* to o^* in $\mathcal{G}_e$. Let $\mathcal{K}_e$ denote the collection of such cuts. To account for the utility cost of removing supporting knowledge, for each supporting triple (s, r, o), we query $\mathcal{M}_{\text{pre}}$ about the triple and obtain its calibrated confidence $u_{\text{pre}}(s, r, o)$ from the model's response. The corresponding fact is annotated as $t = (s, r, o, u)$, where $u = u_{\text{pre}}(s, r, o)$, and assigned a cutting weight $c(t) = u$. This weighting uses confidence as a proxy for removal cost, favoring the preservation of more confidently held facts. Among all feasible cuts, we select the one with minimum total weight, $C_e^* \in \arg\min_{C \in \mathcal{K}_e} \sum_{t \in C} c(t)$, and combine its facts with e to form the final forget set. The resulting cut interrupts all paths from s^* to o^* within the extracted subgraph at minimum total cutting cost.

Discussion. The effectiveness of DUMIC depends on the coverage of its discovered supporting paths. A single search may miss valid paths because explicit responses reveal only part of the model's knowledge. Repeated searches with varied responses can uncover additional paths, but incur further inference and verification costs. To broaden coverage within a limited search budget, DUMIC augments explicitly generated entities with candidates mapped from representations at the model's intermediate layers (Gurnee et al., 2026) during a single search run. The resulting improvement in supporting-path discovery and target coverage is shown in Section 6.2. To reduce redundant computation, we reuse entity expansions across targets and cache direct tail checks for each combination of a target and an intermediate entity. At the final hop, we directly verify connections to the target tail entity, avoiding candidate generation, latent decoding, and relation checks for alternative entities across the final frontier. This eliminates an additional branching factor at the last expansion step, where the frontier can already be large. Concurrent model queries and batched latent decoding further improve throughput. Together, adaptive tree search takes only 80.2 seconds per target on average. Further implementation details and runtime measurements are provided in Appendix D.

5 Evaluation

To rigorously evaluate knowledge discovery and deep unlearning, we require a reference that approximates the model's internal knowledge structure, including the facts it retains and the inferential dependencies among them. Such a reference allows us to assess the validity and coverage of discovered supporting knowledge and identify target facts and their supporting structures for deep unlearning. We use the Harry Potter dataset (Chen et al., 2023) as the starting point for constructing this reference. LLMs may have encountered the novels or related material during pretraining, making it plausible that they retain some of these facts. Relations among characters also provide rich inferential structures for examining whether target knowledge can be reconstructed from supporting facts. As illustrated in Figure 3, we construct and complete a reference knowledge graph from the corpus, then filter its facts using the pretrained model's responses and calibrated confidence.

Constructing the Reference Knowledge Graph. To reduce omissions, knowledge extraction proceeds in two stages: entity extraction and relation extraction. The corpus contains numerous aliases, such as Lily Evans and Lily Potter for the same person. We consolidate canonical names and aliases into a unified entity catalog, preventing different mentions from creating separate nodes across chapters. We then extract relations from each chapter using its text and the entity catalog. To both iden-

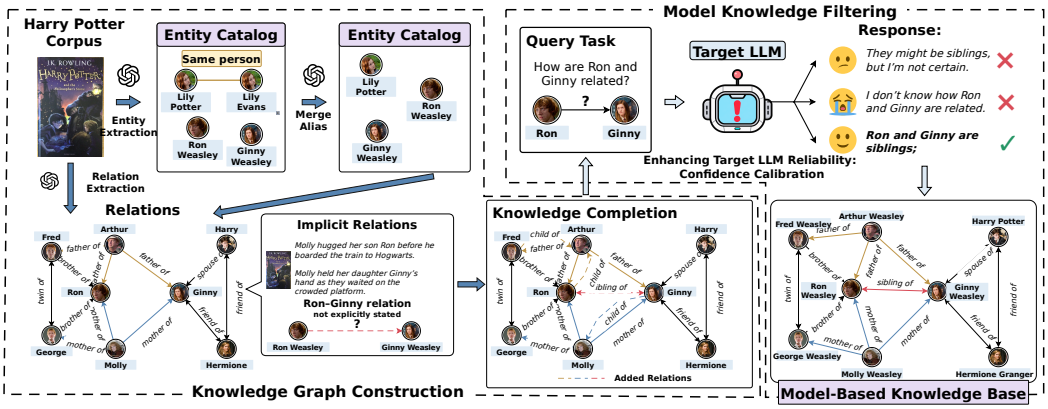

Figure 3: Construction of the model-based knowledge graph. We extract entities and relations from the Harry Potter corpus, merge aliases, and complete relations supported by the extracted facts. We then retain facts recognized by the target LLM according to its calibrated confidence.

tify missing relations and entities, we provide a second model with the same inputs and the initial extraction results for review.

We combine both outputs with their textual evidence and group relations by subject and object. Equivalent or overlapping descriptions, such as SIBLINGOF and SISTEROF for the same entity pair, are consolidated by retaining the more specific relation supported by the evidence. To recover relations left implicit in the text, we examine each entity's neighborhood and supplement missing inverse relations and relations implied between neighboring entities. We record the supporting facts for each addition. These steps produce the reference KG $\mathcal{G}_{\text{ref}}$.

Filtering Facts According to Model Knowledge. We express each candidate triple as a multiple choice question with options A, B, C, D, and Unknown. We query the pretrained model $\mathcal{M}_{\text{pre}}$ and obtain confidence u from its calibrated answer probabilities. A fact passes the knowledge filter only when the model selects the correct answer and satisfies the confidence criterion. We apply the same probing procedure to target triples and candidate supporting facts, and use this procedure consistently during reference construction and evaluation after unlearning.

For each pretrained model, we first identify candidate target triples that pass the knowledge filter. Given a candidate $e = (s^*, r^*, o^*)$, we extract candidate paths connecting s^* and o^* in $\mathcal{G}_{\text{ref}}$, excluding the target fact itself from the supporting evidence. For each path, we provide the target triple and the constituent facts to the judge, which determines whether these facts jointly support the inference of e. The judgment is restricted to the supplied facts, and paths that fail this assessment are removed. We then apply the knowledge filter to the constituent facts of each remaining path. A path survives only when all its facts pass this filter.

6 EXPERIMENTS

6.1 EXPERIMENTAL SETTINGS

Experimental Setup We conduct experiments on Qwen3.5-27B (Qwen Team, 2026) and Gemma4-31B-Base (Gemma Team, 2026), with a training budget of five epochs. We consider three unlearning settings: *Superficial Unlearning*, which removes only the target facts; *Deep Unlearning*, which removes the target facts and all facts in their discovered supporting subgraphs; and *DUMIC Unlearning*, which removes the target facts and supporting facts selected by the confidence-aware minimum cut. We set the learning rate to 5×10^{-6} for full parameter fine-tuning and 1×10^{-5} for LoRA. For epoch selection, we select the last epoch whose accuracy on the utility set used for epoch selection exceeds 0.8; if no epoch satisfies this criterion, we select the epoch with the highest accuracy. We use GPT-5 (OpenAI, 2025) and Gemini 3.1 Pro (Google DeepMind, 2026) to construct the reference knowledge graph. Gemini 3.1 Pro is additionally used for supporting-path search and evaluation dataset construction. We use the Jacobian lens (J-lens) (Gurnee et al., 2026) for latent entity discovery during search. Detailed implementation settings are provided in Appendix A.

Table 1: Comparison of different unlearning settings on Qwen3.5-27B and Gemma4-31B-Base. Percentages indicate relative UES improvements over superficial unlearning.

Method	Setting	Qwen3.5-27B				Gemma4-31B-Base			
		Unlearning Effectiveness	Utility Retention			Unlearning Effectiveness	Utility Retention		
		UES ↑	Loc ↑	Gen ↑	Rea ↑	UES ↑	Loc ↑	Gen ↑	Rea ↑
No Unlearn		0.000	1.000	0.818	0.900	0.000	1.000	0.760	0.814
GA	Superficial Unlearn	0.799	0.885	0.816	0.894	0.552	0.888	0.748	0.740
	DeepUnlearn	0.963	0.707	0.814	0.874	1.000	0.000	0.738	0.000
	DUMIC	0.926 (15.89%↑)	0.745	0.812	0.886	0.690 (25.00%↑)	0.905	0.756	0.698
NPO	Superficial Unlearn	0.794	0.889	0.812	0.896	0.568	0.897	0.748	0.746
	DeepUnlearn	0.965	0.647	0.814	0.878	1.000	0.000	0.732	0.000
	DUMIC	0.926 (16.62%↑)	0.777	0.808	0.882	0.700 (23.24%↑)	0.900	0.742	0.712
SimNPO	Superficial Unlearn	0.806	0.854	0.820	0.894	0.419	0.898	0.746	0.788
	DeepUnlearn	0.946	0.652	0.806	0.886	0.809	0.790	0.730	0.706
	DUMIC	0.905 (12.28%↑)	0.780	0.816	0.904	0.687 (63.96%↑)	0.905	0.746	0.738
$\mathrm{GA}_{\mathrm{GDR}}$	Superficial Unlearn	0.839	0.873	0.812	0.896	0.107	0.976	0.758	0.744
	DeepUnlearn	0.937	0.742	0.808	0.876	0.208	0.940	0.754	0.106
	DUMIC	0.936 (11.56%↑)	0.781	0.818	0.880	0.491 (358.88%↑)	0.948	0.742	0.590
$\mathrm{NPO}_{\mathrm{GDR}}$	Superficial Unlearn	0.555	0.948	0.816	0.880	0.387	0.926	0.762	0.778
	DeepUnlearn	0.947	0.613	0.820	0.882	0.999	0.196	0.748	0.662
	DUMIC	0.883 (59.10%↑)	0.891	0.816	0.886	0.583 (50.65%↑)	0.844	0.758	0.744

Unlearning Methods For each target LLM, we evaluate five representative unlearning approaches. Gradient Ascent (GA) Jang et al. (2023) maximizes the prediction loss on the forget set to reduce the likelihood of the original responses. Negative Preference Optimization (NPO) Zhang et al. (2024) treats responses in the forget set as negative examples and reduces their likelihood relative to a frozen pretrained reference model. $\mathrm{GA}_{\mathrm{GDR}}$ Yao et al. (2024a) and $\mathrm{NPO}_{\mathrm{GDR}}$ Zhang et al. (2024) combine their respective forgetting objectives with gradient descent on the prediction loss of the retain set to preserve model utility. Finally, Simple Negative Preference Optimization (SimNPO) Fan et al. (2025) uses a negative preference objective based on response log probabilities normalized by length, without requiring a reference model.

Evaluation Metrics. We evaluate deep unlearning with the Unlearning Effectiveness Score (UES). For a supporting fact t, let $C_m(t)$ be its calibrated confidence under model m; a path P has confidence $C_m(P) = \min_{t \in P} C_m(t)$. Let $I(P)$ indicate whether any fact on P receives zero confidence or an incorrect answer. We define $s(P) = I(P) + (1 - I(P))[1 - C_{\mathrm{unl}}(P)/C_{\mathrm{pre}}(P)]$ and $\mathrm{UES} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{|\mathcal{P}_i|} \sum_{P \in \mathcal{P}_i} s(P)$, where N is the number of targets and $\mathcal{P}_i$ is the set of evaluated supporting paths for target i. For local utility, $\mathcal{N}_i$ contains 30 facts sampled within three hops of the target's subject or object, excluding the target and its supporting subgraph (Wei et al., 2025). We define $\mathrm{Loc} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{|\mathcal{N}_i|} \sum_{t \in \mathcal{N}_i} \mathbb{I}[\widehat{y}_{\mathrm{pre}}(t) = \widehat{y}_{\mathrm{unl}}(t)]$, where $\widehat{y}_m(t)$ is model m's predicted answer. Gen and Rea assess general knowledge on MMLU (Hendrycks et al., 2021) and reasoning on BBH (Suzgun et al., 2023), respectively. See Appendix A.5 for details.

Datasets For each target model, we select the 100 recognized triples with the most verified supporting paths in the reference knowledge graph as unlearning targets. We then independently discover a model-specific supporting subgraph for each target using the search procedure in Section 4. We consider three forget sets: *Unlearn* contains only the targets; *DeepUnlearn* contains the targets and all facts in their searched supporting subgraphs; and *DUMIC* contains the targets and the supporting facts selected by the minimum cut. We convert reference supporting facts and three-hop neighboring facts into A/B/C/D/Unknown multiple-choice questions with balanced gold-answer positions. Further details are provided in Appendix B.

6.2 Main Results

Deep Unlearning Strengthens Forgetting Table 1 shows that DeepUnlearn consistently improves UES over superficial unlearning across both models. For example, on Gemma-4-31B-Base, $\mathrm{NPO}_{\mathrm{GDR}}$ improves from 0.387 to 0.999. On Qwen3.5-27B, DeepUnlearn achieves UES above 0.93 across all five methods, indicating that explicitly targeting supporting facts weakens recovery

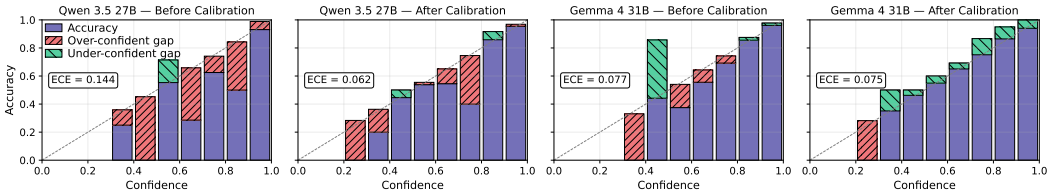

Figure 5: Reliability diagrams before and after temperature scaling.

paths more effectively than unlearning the targets alone. However, removing the entire supporting subgraph can severely damage utility: on Gemma-4-31B-Base, $\mathrm{NPO}_{\mathrm{GDR}}$ reduces Loc from 0.926 to 0.196, while GA and NPO reduce both Loc and Rea to zero. These results show that removing supporting knowledge strengthens forgetting but may incur substantial utility loss.

Utility Preservation through Minimum-Cut Selection Minimum-cut selection mitigates this utility loss while maintaining higher UES than superficial unlearning across all five methods on both models. On Gemma-4-31B-Base, it restores GA's Loc and Rea from zero under DeepUnlearn to 0.905 and 0.698, respectively. For $\mathrm{NPO}_{\mathrm{GDR}}$, it improves Loc from 0.196 to 0.844, while achieving a UES of 0.583, compared with 0.387 under superficial unlearning. Methods with lower superficial UES often obtain larger gains, potentially because explicitly targeting supporting facts addresses recovery paths left largely intact by target-only unlearning. Lower baselines also leave more room for improvement and amplify relative percentage gains.

Effectiveness of Supporting Path Discovery As shown in Figure 4, J-Lens discovers 2,632 candidate paths on Gemma and 3,343 on Qwen, of which 1,376 and 2,618 remain after filtering, respectively. These retained paths cover 93% of targets on Gemma and 96% on Qwen, showing that the procedure identifies supporting paths for most targets. For comparison, searching the reference KG yields 2,794 retained paths on Gemma and 3,197 on Qwen, covering all targets on both models. Although model-guided search does not exhaust the supporting paths in the reference KG, J-Lens discovers substantial supporting evidence without using the reference KG to guide its search.

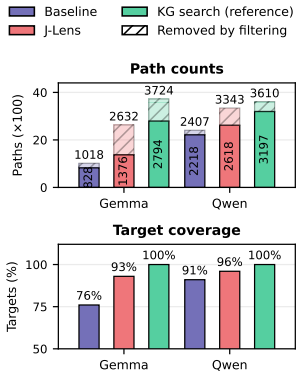

Figure 4: Supporting path discovery. Top: path counts before and after filtering, with hatched segments indicating removed paths. Bottom: target coverage after filtering.

Contribution of the Jacobian Lens We next compare J-Lens with the baseline to assess the contribution of the Jacobian lens. Before filtering, J-Lens increases the number of discovered paths from 1,018 to 2,632 on Gemma and from 2,407 to 3,343 on Qwen, corresponding to 2.59× and 1.39× the baseline counts, respectively. The gains persist after filtering: the number of retained paths increases from 828 to 1,376 on Gemma (+66.2%) and from 2,218 to 2,618 on Qwen (+18.0%). Target coverage also improves, rising from 76% to 93% on Gemma and from 91% to 96% on Qwen. These improvements show that the benefits of J-Lens extend to both the number of retained supporting paths and the fraction of targets covered.

6.3 Reliability of Model Confidence Estimates

To assess the reliability of model confidence for knowledge evaluation, we apply temperature scaling to the answer token probabilities. Using a calibration set containing 200 positive and 200 negative triples, we fit temperatures of 1.6 for Qwen3.5-27B and 1.3 for Gemma4-31B-Base. We evaluate calibration on a separate set of 100 positive and 100 negative triples using expected calibration error and reliability diagrams. As illustrated in Figure 5, calibration reduces the expected calibration error of Qwen3.5-27B from 0.144 to 0.062, indicating improved alignment between predicted confidence and observed accuracy. Gemma4-31B-Base exhibits a lower initial calibration error, with a modest reduction from 0.077 to 0.075. We subsequently use the calibrated probabilities to compute entropy for knowledge filtering and supporting subgraph evaluation.

Table 2: Comparison of unlearning methods using LoRA on Qwen3.5-27B under superficial unlearning and deep unlearning with minimum cut.

Method	Superficial Unlearning				DUMIC			
	Unlearning Effectiveness	Utility Retention			Unlearning Effectiveness	Utility Retention		
	UES ↑	Loc ↑	Gen ↑	Rea ↑	UES ↑	Loc ↑	Gen ↑	Rea ↑
Base Model	0.000	1.000	0.818	0.900	0.000	1.000	0.818	0.900
GA	0.109	0.923	0.812	0.834	0.789 (623.85%↑)	0.730	0.802	0.622
NPO	-0.168	0.935	0.812	0.846	0.337 (300.60%↑)	0.914	0.812	0.776
SimNPO	0.083	0.924	0.818	0.894	0.141 (69.88%↑)	0.944	0.812	0.886
GA_{GDR}	-0.643	0.936	0.808	0.706	0.548 (185.23%↑)	0.911	0.806	0.686
NPO_{GDR}	-0.377	0.962	0.816	0.864	-0.051 (86.47%↑)	0.957	0.816	0.882

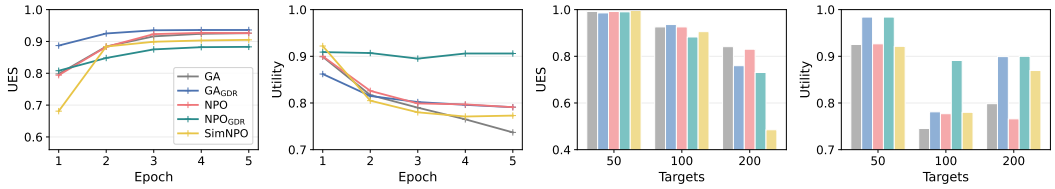

Figure 6: Effects of unlearning epochs (left) and target count (right) on UES and utility for Qwen3.5-27B with full-parameter fine-tuning and DUMIC.

6.4 Ablation Studies

Effectiveness of Deep Unlearning under Low-Rank Adaptation Table 2 shows that DUMIC improves UES across all five methods under LoRA. The gains are particularly large for GA and GA_{GDR}: GA improves from 0.109 to 0.789, while GA_{GDR} improves from -0.643 to 0.548. GA_{GDR} achieves these gains with modest changes in Loc, Gen, and Rea, whereas GA incurs larger losses in local utility and reasoning performance. These results demonstrate that our framework also applies to LoRA, improving unlearning effectiveness under parameter-efficient adaptation.

Impact of Unlearning Iterations We analyze five unlearning epochs on Qwen3.5-27B under DUMIC. As shown in Figure 6, UES improves mainly during the first two to three epochs and then largely stabilizes. SimNPO exhibits the largest early gain, with UES increasing from approximately 0.68 to 0.88 between the first and second epochs. Local utility generally declines with further training; for example, GA's Loc decreases from approximately 0.92 to 0.74. These results indicate diminishing forgetting gains in later epochs, often accompanied by additional utility loss.

Impact of Target Count The right two panels of Figure 6 compare unlearning the top 50, 100, and 200 targets, ranked by the number of verified supporting paths in the reference knowledge graph. Average UES decreases from 0.9913 to 0.9152 and 0.7294, respectively. This decline appears across all five methods: from 50 to 200 targets, GA falls from 0.9921 to 0.8412, while SimNPO falls from 0.9962 to 0.4852. Thus, forgetting becomes less effective as the target set expands. This decline may partly reflect lower supporting-path coverage for the additional targets, leaving more recovery paths outside the discovered subgraphs and therefore beyond the reach of minimum-cut selection.

7 Conclusion

We introduced a deep unlearning framework that discovers model-specific supporting paths through adaptive tree search and latent entity extraction from intermediate representations. After verifying these paths and constructing a supporting subgraph, the framework uses a confidence-aware minimum cut to select facts for unlearning while limiting collateral knowledge loss. Experiments on two LLMs show improved path coverage and a better balance between unlearning effectiveness and utility preservation. These findings highlight the importance of addressing related knowledge that can reconstruct a target fact.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and grammar correction. All research decisions, experimental analyses, and final manuscript content were reviewed and verified by the authors, who take full responsibility for the work.

ETHICS STATEMENT

This work studies knowledge unlearning in large language models, with the goal of understanding how target knowledge can remain recoverable through related information after unlearning. Our experiments use a fictional knowledge domain and do not involve human participants or personal or sensitive user data. The study is conducted in a controlled research setting and does not involve deployment in real-world systems. More broadly, the techniques studied here are intended to improve the reliability of machine unlearning and the removal of undesired model knowledge.

REPRODUCIBILITY STATEMENT

The main text and appendix provide the methodological, dataset, training, and evaluation details needed to reproduce the experiments in this work. In particular, we describe the supporting-path search procedure, latent entity discovery, confidence calibration, unlearning configurations, dataset construction, evaluation metrics, and compute settings, together with additional ablations and efficiency analyses.

REFERENCES

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pp. 463–474. IEEE, 2015. doi: 10.1109/SP.2015.35.

Nuo Chen, Yan Wang, Haiyun Jiang, Deng Cai, Yuhan Li, Ziyang Chen, Longyue Wang, and Jia Li. Large language models meet Harry Potter: A dataset for aligning dialogue agents with characters. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 8506–8520. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.570. URL https://aclanthology.org/2023.findings-emnlp.570/.

Keyuan Cheng, Zijian Kan, Zhuoran Zhang, Muhammad Asif Ali, Lijie Hu, and Di Wang. COMPKE: Complex question answering under knowledge editing. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 2557–2576. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.130. URL https://aclanthology.org/2025.findings-acl.130/.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298, 2024. doi: 10.1162/tacl_a_00644. URL https://aclanthology.org/2024.tacl-1.16/.

Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023. URL https://arxiv.org/abs/2310.02238.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0058. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/02443e4e008231e0af0855f3cc70ed17-Abstract-Conference.html.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Google DeepMind. Gemini 3.1 Pro model card, February 2026. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-1-Pro-Model-Card.pdf.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. *arXiv preprint arXiv:2607.15495*, 2026. URL https://arxiv.org/abs/2607.15495.

Peter Henderson, Xuechen Li, Dan Jurafsky, Tatsunori Hashimoto, Mark A. Lemley, and Percy Liang. Foundation models and fair use. *Journal of Machine Learning Research*, 24(400):1–79, 2023. URL https://jmlr.org/papers/v24/23-0569.html.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

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*, pp. 14389–14408. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.805. URL https://aclanthology.org/2023.acl-long.805/.

Jinghan Jia, Yihua Zhang, Yimeng Zhang, Jiancheng Liu, Bharat Runwal, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. SOUL: Unlocking the power of second-order optimization for LLM unlearning. *arXiv preprint arXiv:2404.18239*, 2024. URL https://arxiv.org/abs/2404.18239.

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. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3117. URL https://papers.nips.cc/paper/2024/hash/b1f78dfc9ca0156498241012aec4efa0-Abstract-Datasets_and_Benchmarks_Track.html.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/062d711fb777322e2152435459e6e9d9-Abstract-Conference.html.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. Rethinking machine unlearning for large language models. *arXiv preprint arXiv:2402.08787*, 2024. URL https://arxiv.org/abs/2402.08787.

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 *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5345–5363. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.coling-main.358/.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J. Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *ICLR 2024 Workshop on Data Problems for Foundation Models*, 2024. URL https://openreview.net/forum?id=BDFi8mQWTY.

Sewon Min, Suchin Gururangan, Eric Wallace, Weijia Shi, Hannaneh Hajishirzi, Noah A. Smith, and Luke Zettlemoyer. SILO language models: Isolating legal risk in a nonparametric datastore. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ruk0nyQPec.

Peizhi Niu, Evelyn Ma, Huiting Zhou, Duo Zhou, Huan Zhang, S Rasoul Etesami, and Olgica Milenkovic. Guard: Guided unlearning and retention via data attribution for large language models. *arXiv preprint arXiv:2506.10946*, 2025.

OpenAI. GPT-5 system card, August 2025. URL https://openai.com/index/gpt-5-system-card/.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16, 2020. doi: 10.1109/SC41405.2020.00024.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18075–18086, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9627c45df543c816a3ddf2d8ea686a99-Abstract.html.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. MUSE: Machine unlearning six-way evaluation for language models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TArmA033BU.

Yucheng Shi, Qiaoyu Tan, Xuansheng Wu, Shaochen Zhong, Kaixiong Zhou, and Ninghao Liu. Retrieval-enhanced knowledge editing in language models for multi-hop question answering. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2056–2066, 2024.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

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. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 1962–1994. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-long.88. URL https://aclanthology.org/2026.acl-long.88/.

Jialu Wang, Jianing Deng, Shuqing Luo, Yuanzhe Li, Dongwei Wang, Jingtong Hu, Huanrui Yang, Song Wang, and Tianlong Chen. Quantization-robust unlearning through the lens of retain-forget loss landscapes interaction, 2026b. URL https://arxiv.org/abs/2609.27355.

Rongzhe Wei, Mufei Li, Mohsen Ghassemi, Eleonora Kreačić, Yifan Li, Xiang Yue, Bo Li, Vamsi K Potluru, Pan Li, and Eli Chien. Underestimated privacy risks for minority populations in large language model unlearning. *arXiv preprint arXiv:2412.08559*, 2024.

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. Do LLMs really forget? evaluating unlearning with knowledge correlation and confidence awareness. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/85ec26ef94c3acb4c195e905df1ff4f7-Paper-Conference.pdf.

Ruihan Wu, Chhavi Yadav, Russ Salakhutdinov, and Kamalika Chaudhuri. Evaluating deep unlearning in large language models. *arXiv preprint arXiv:2410.15153*, 2024. URL https://arxiv.org/abs/2410.15153.

Ruihan Wu, Konstantin Garov, and Kamalika Chaudhuri. Learning-time encoding shapes unlearning in llms, 2025. URL https://arxiv.org/abs/2506.15076.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. Machine unlearning of pre-trained large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 8403–8419. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.acl-long.457. URL https://aclanthology.org/2024.acl-long.457/.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. In *Advances in Neural Information Processing Systems*, volume 37, 2024b. doi: 10.52202/079017-3346. URL https://papers.nips.cc/paper/2024/hash/be52acf6bccf4a8c0a90fe2f5cfcead3-Abstract-Conference.html.

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

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15686–15702. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.971. URL https://aclanthology.org/2023.emnlp-main.971/.

SUPPLEMENTARY MATERIAL

Appendix

Contents

A Detailed Experimental Settings **15**
A.1 Compute Configuration 15
A.2 Search Configuration 15
A.3 J-Lens Configuration 15
A.4 Training Configuration 15
A.5 Detailed Definitions of Evaluation Metrics 15

B Dataset Construction Details **16**

C Knowledge Graph Statistics **17**

D Efficiency of Supporting Path Discovery **17**

E Mapping Between Entropy Threshold and Correct-Answer Probability **18**

F Additional Ablation Studies **18**
F.1 Controlling for Forget-Set Size 18

A Detailed Experimental Settings

A.1 Compute Configuration

All experiments were conducted on NVIDIA A100 80GB PCIe GPUs, which were used for both training and inference.

A.2 Search Configuration

We use `vLLM serve` (Kwon et al., 2023) for efficient model inference during supporting-path search, with batch sizes of 512 for Qwen3.5-27B and 256 for Gemma4-31B-Base. We compare two search settings in Figure 4: the baseline uses only entities explicitly generated in model responses, whereas J-Lens additionally incorporates latent entities decoded from intermediate representations.

A.3 J-Lens Configuration

We apply J-Lens (Gurnee et al., 2026) to token positions within the first three object fields of each probe response, using only the last of the three inspected layers for candidate extraction. Within each field, we aggregate readout probabilities by taking the maximum across its token positions. To reduce noise from spurious token fragments, we retain up to three new latent entities per field with readout probabilities of at least 10^{-4}. Each expansion admits at most 50 candidates, with explicitly generated entities taking priority over latent candidates.

A.4 Training Configuration

All unlearning experiments were performed using four GPUs with DeepSpeed ZeRO Stage 3 (Rajbhandari et al., 2020). The per-GPU batch size was set to 4. Selected training epochs for all combinations of models, methods, and unlearning settings are reported in Table 3.

Table 3: Selected training epochs under different unlearning settings using full-parameter fine-tuning (Full) and LoRA. Sup., Deep, and +Cut denote superficial unlearning, deep unlearning, and deep unlearning with minimum cut.

Method	**Qwen3.5-27B**						**Gemma4-31B-Base**		
	Full			LoRA			Full		
	Sup.	Deep	+Cut	Sup.	Deep	+Cut	Sup.	Deep	+Cut
GA	5	5	5	5	5	5	2	1	1
NPO	5	5	5	5	5	5	2	1	1
SimNPO	5	5	5	5	5	5	5	5	5
GA_{GDR}	5	5	5	5	1	5	1	1	1
NPO_{GDR}	5	5	5	5	5	5	5	5	5

A.5 Detailed Definitions of Evaluation Metrics

Let $\mathcal{D}_{\text{forget}} = \{e_i\}_{i=1}^{N}$ denote the set of target triples. For each target e_i, let $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ denote its supporting subgraph, where $\mathcal{E}_i$ contains the unique facts in the union of all verified supporting paths. We use $\mathcal{P}_i$ to denote the set of supporting paths for e_i. The pretrained and unlearned models are denoted by $\mathcal{M}_{\text{pre}}$ and $\mathcal{M}_{\text{unl}}$, respectively.

Unlearning Effectiveness Score (UES). UES measures the reduction in confidence of supporting paths after unlearning. For each supporting fact t, let $C_{\text{pre}}(t)$ and $C_{\text{unl}}(t)$ denote its calibrated confidence under the pretrained and unlearned models, respectively. Since reconstructing the target through a path requires all its constituent facts, we approximate path confidence by the lowest confidence among these facts:

$$C_m(P) = \min_{t \in P} C_m(t), \qquad m \in \{\text{pre}, \text{unl}\}. \tag{1}$$

We consider a path interrupted if, after unlearning, at least one constituent fact has zero confidence or is answered incorrectly, including an Unknown response. Such a path receives a score of 1. Otherwise, its score is the normalized reduction in path confidence:

$$s(P) = \begin{cases} 1, & \text{if } \exists\, t \in P : C_{\text{unl}}(t) = 0 \text{ or } \hat{a}_{\text{unl}}(t) \neq a^*(t), \\ \dfrac{C_{\text{pre}}(P) - C_{\text{unl}}(P)}{C_{\text{pre}}(P)}, & \text{otherwise}, \end{cases} \tag{2}$$

where $\hat{a}_{\text{unl}}(t)$ and $a^*(t)$ denote the unlearned model's predicted answer and the correct answer for fact t, respectively. We evaluate paths with positive pretrained confidence. UES averages these scores over supporting paths and then over targets:

$$\text{UES} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{P}_i|} \sum_{P \in \mathcal{P}_i} s(P). \tag{3}$$

A higher UES indicates more effective interruption or weakening of supporting paths. A value of 1 indicates that every evaluated path is interrupted under the above criterion, while a negative value indicates that confidence increases in the remaining paths outweigh the contributions from interrupted or weakened paths.

Local Consistency (Loc). Loc evaluates whether unlearning preserves neighboring factual knowledge that is not involved in reconstructing the target. For each target $e_i = (s_i, r_i, o_i)$, we sample 30 facts from the three-hop neighborhoods of s_i and o_i in the reference knowledge graph. We exclude the target itself and all facts in its supporting subgraph. Let $\mathcal{N}_i$ denote the resulting local utility set, and let $\widehat{y}_{\mathcal{M}}(t)$ denote the model's prediction among $\{\text{A}, \text{B}, \text{C}, \text{D}, \textsc{Unknown}\}$. Loc is defined as

$$\text{Loc} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{N}_i|} \sum_{t \in \mathcal{N}_i} \mathbb{I}\left[\widehat{y}_{\text{pre}}(t) = \widehat{y}_{\text{unl}}(t)\right]. \tag{4}$$

A higher Loc indicates better preservation of local factual knowledge around the unlearning targets.

General Knowledge and Reasoning Ability. General Knowledge (Gen) is measured on MMLU (Hendrycks et al., 2021) using 5-shot perplexity-based answer ranking. Reasoning Ability (Rea) is measured on the 27 tasks in BBH (Suzgun et al., 2023) using 3-shot chain-of-thought prompting and exact-match accuracy. Higher Gen and Rea indicate better preservation of general knowledge and reasoning capabilities, respectively.

B Dataset Construction Details

Unlearning Targets and Supporting Subgraphs. We select the 100 target triples with the largest numbers of verified supporting paths in the reference knowledge graph constructed in Section 5. We define 19 sentence templates and randomly sample one for each target triple, converting it into a natural-language sentence for unlearning. For each target e_i, the discovered paths and their facts form a model-specific searched subgraph $\widehat{\mathcal{G}}_i$, which is used to construct the forget sets and perform deep unlearning. The resulting forget-set sizes and retain sets are summarized in Table 4. We further retrieve the corresponding supporting paths from the reference knowledge graph and combine them into a reference subgraph $\mathcal{G}_i^{\text{ref}}$. This fixed, method-independent subgraph is used to measure fact-level and path-level unlearning effectiveness.

Multiple-Choice Evaluation. We prompt an LLM to convert both the facts in the reference supporting subgraphs and the local utility facts into four-option multiple-choice questions. Each question contains one gold answer and three plausible distractors, with gold positions evenly balanced across A–D. The supporting-subgraph questions are used to compute UES. For Loc, we sample 30 facts from the three-hop neighborhoods of each target's head and tail entities, excluding the target and its supporting facts. We compare model predictions on these fixed questions before and after unlearning. For epoch selection, we randomly sample 100 questions from a separate utility set.

Table 4: Numbers of forget and retain examples used under different unlearning settings. Each retain set is matched in size to its corresponding forget set.

Model	Unlearn		DUMIC		DeepUnlearn	
	Forget	Retain	Forget	Retain	Forget	Retain
Qwen3.5-27B	100	100	236	236	616	616
Gemma4-31B-Base	100	100	232	232	639	639

Confidence Calibration Dataset. We construct separate calibration sets for each target model. The temperature-fitting set contains 200 positive and 200 negative examples, while the held-out calibration evaluation set contains another 100 positive and 100 negative examples. All examples are presented using the same A/B/C/D/Unknown format as the effectiveness and utility evaluations. We fit a separate temperature for each model on the fitting set and report calibration quality on the held-out set. The calibration examples are separate from the unlearning targets and their effectiveness and utility evaluation questions.

C Knowledge Graph Statistics

The Harry Potter series provides a coherent fictional world with rich character relationships and multi-hop factual dependencies. We use the complete English texts of all seven original Harry Potter books as our corpus, following the fictional domain studied by the Harry Potter Dialogue dataset, a bilingual collection of character-centered dialogues (Chen et al., 2023).

Figure 7 summarizes the numbers of entities and relation triples extracted from each of the seven Harry Potter books. The per-book extractions contain between 179 and 386 entities and between 510 and 1,174 relation triples, with Book 5 yielding the largest counts in both categories. After consolidating the extracted knowledge and removing duplicates, the resulting reference knowledge graph contains 1,862 entities and 4,122 relation triples.

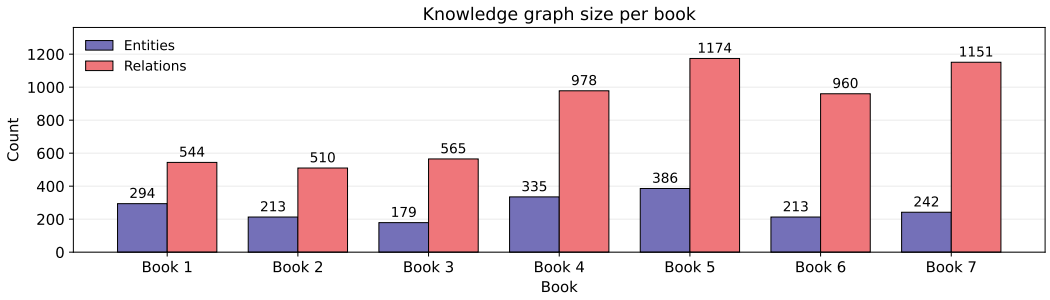

Figure 7: Numbers of entities and relation triples extracted from each of the seven Harry Potter books.

D Efficiency of Supporting Path Discovery

We use vLLM (Kwon et al., 2023) for model inference during supporting-path search. Table 5 reports the runtime and path statistics for 200 target facts. On Qwen3.5-27B, our framework requires an average of 80.2 seconds per target, discovering 54.9 candidate paths and retaining 20.3 accepted paths per target. On Gemma4-31B-Base, the corresponding runtime is 157.9 seconds, with averages of 46.9 candidate paths and 15.9 accepted paths per target. Thus, the average search time remains below three minutes per target on both models at the evaluated scale.

Table 5: Runtime and path statistics of supporting-path discovery over 200 target facts. Candidate and accepted path counts are totals across all targets.

Model	Total time	Avg. per target	Candidate paths	Accepted paths
Qwen3.5-27B	4.45 h	80.2 s	10,971	4,067
Gemma4-31B-Base	8.77 h	157.9 s	9,370	3,181

Table 6: Feasible correct-answer probability ranges at fixed entropy values $H(\mathbf{q}) = u^*$ for five options (A, B, C, D, and Unknown), assuming the correct option has the highest probability. Entropy is measured in bits.

Entropy	Correct Prob. Range	Entropy	Correct Prob. Range
0.10	$[0.987, 0.990]$	0.60	$[0.854, 0.913]$
0.15	$[0.978, 0.984]$	0.65	$[0.833, 0.904]$
0.20	$[0.969, 0.978]$	0.70	$[0.811, 0.894]$
0.25	$[0.958, 0.971]$	0.75	$[0.785, 0.884]$
0.30	$[0.947, 0.963]$	0.80	$[0.757, 0.874]$
0.35	$[0.934, 0.956]$	0.85	$[0.724, 0.863]$
0.40	$[0.921, 0.948]$	0.90	$[0.684, 0.852]$
0.45	$[0.906, 0.940]$	0.95	$[0.631, 0.841]$
0.50	$[0.890, 0.931]$	1.00	$[0.500, 0.829]$
0.55	$[0.873, 0.922]$		

E Mapping Between Entropy Threshold and Correct-Answer Probability

In this section, we explain how the entropy-based filtering criterion used for constructing supporting subgraphs relates to the model's probability of selecting the correct answer. We consider five options, $\mathcal{A} = \{A, B, C, D, \text{Unknown}\}$, and denote the correct option by $a^* \in \{A, B, C, D\}$. Let $\mathbf{q}$ be the model's probability distribution normalized over these five options. A prediction is considered to reflect retained knowledge when (i) the correct option has the highest probability and (ii) the entropy of $\mathbf{q}$ is below a threshold u^*. We compute entropy in bits as $H(\mathbf{q}) = -\sum_{a \in \mathcal{A}} q_a \log_2 q_a$.

Table 6 reports the feasible ranges of the correct-answer probability $p = q_{a^*}$ at different fixed entropy values $H(\mathbf{q}) = u^*$, subject to the correct option being an argmax. The upper bound occurs when the other four options share the remaining probability equally:

$$u^* = H\left(p, \frac{1-p}{4}, \frac{1-p}{4}, \frac{1-p}{4}, \frac{1-p}{4}\right), \qquad p \geq \frac{1-p}{4}. \tag{5}$$

For the range $0 \leq u^* \leq 1$ bit considered here, the lower bound occurs when one alternative receives all the remaining probability and the other three receive zero:

$$u^* = H(p, 1-p, 0, 0, 0), \qquad p \geq 1-p. \tag{6}$$

These intervals characterize the boundary $H(\mathbf{q}) = u^*$. Under the actual filtering condition $H(\mathbf{q}) < u^*$, the correct-answer probability exceeds the corresponding lower bound and may reach 1.

F Additional Ablation Studies

F.1 Controlling for Forget-Set Size

To determine whether the gains of minimum-cut selection arise merely from using more training examples, we expand each superficial forget set to match the corresponding minimum-cut set of 236 examples and train both settings for five epochs. As shown in Table 7, DUMIC achieves higher UES across all five methods, with absolute gains ranging from 0.033 to 0.167, although Loc decreases in every case. The largest UES gain occurs for NPO_{GDR}, whose UES increases from 0.716 to 0.883 while Loc decreases from 0.924 to 0.891. These results show that the forgetting gains cannot be explained by forget-set size alone: selecting supporting facts improves unlearning effectiveness under the same example budget, at the cost of some local utility.

Table 7: Superficial unlearning versus DUMIC on Qwen3.5-27B with 236 forget examples and five training epochs. Higher UES is bolded.

Method	**UES** ↑		**Loc** ↑	
	Superficial	DUMIC	Superficial	DUMIC
GA	0.871	**0.926**	0.857	0.745
$\mathrm{GA}_{\mathrm{GDR}}$	0.871	**0.936**	0.863	0.781
NPO	0.893	**0.926**	0.870	0.777
$\mathrm{NPO}_{\mathrm{GDR}}$	0.716	**0.883**	0.924	0.891
SimNPO	0.869	**0.905**	0.817	0.780