Rethinking LLM Unlearning: From Safety Constraints to Functional Utility

Anonymous EMNLP submission

Abstract

Machine unlearning aims to approximate the counterfactual LLM that would have been trained without specified data. Given its rapid development, a timely and comprehensive review is essential. Despite the emergence of surveys on LLM unlearning, existing reviews lag behind rapid progress in benchmarks, unlearning algorithms, and evaluation under safety objectives. Moreover, they largely overlook functional uses. We address these gaps by offering an up-to-date, application-oriented survey that organizes the literature into *safety forgetting* and *functional forgetting*. For safety forgetting, we revisit benchmarks, unlearning algorithms, and evaluation, highlighting recent developments such as automatically synthesized benchmarks, the distinction between single and continual settings, and evaluation challenges for reasoning LLMs. Beyond safety forgetting, unlearning may also serve as a functional tool. We thus identify functional forgetting as an important yet underexplored direction, spanning behavior probing, graceful forgetting of outdated knowledge, and provide a case study for model regularization. We conclude with open challenges and directions toward a unified understanding of safety and functional forgetting.

1 Introduction

Large language models (LLMs) trained on massive corpora achieve strong performance in mathematics and code generation. However, their ability to memorize portions of the training data raises ethical, legal, and security concerns. Prior work shows that LLMs can reproduce personal information, harmful content, or copyright (Karamolegkou et al., 2023). Moreover, regulations such as the EU's Right to be Forgotten (Ginart et al., 2019) underscore the need for mechanisms that enforce data removal. Motivated by these concerns, *machine unlearning* aims to update a trained model so that it behaves as if specific training examples had never been learned (Cao and Yang, 2015).

Machine unlearning for LLMs has attracted increasing attention in recent years, primarily as a data removal mechanism for safety objectives. Meanwhile, LLMs are rapidly evolving in scale and capability, driven by larger model capacity and new training paradigms. This co-evolution introduces new challenges and complexities for unlearning, such as potential leakage arising from thinking behaviors (Wang et al., 2025a). Beyond data removal, unlearning may also serve as a functional tool. For example, prior work suggests that unlearning followed by relearning can improve model accuracy, indicating its potential to enhance model capabilities (Xu et al., 2025b). However, this functional perspective remains relatively underexplored.

While several preliminary surveys on LLM unlearning have emerged (Liu et al., 2025; Qiu et al., 2025; Blanco-Justicia et al., 2025; Le-Khac and Truong, 2025; Ren et al., 2025b; Geng et al., 2025), they still lag behind recent developments in safety-oriented data removal and largely overlook functional uses. As shown in Figure 1 and summarized in Table 1, we provide an application-oriented and up-to-date view of the field by organizing LLM unlearning into two categories based on application focus: *safety forgetting*, which is driven by safety objectives and covers privacy, copyright, and harmful content, and *functional forgetting*, which uses unlearning to analyze or improve model behavior, including behavior probing, graceful forgetting of outdated knowledge, and model regularization to reduce overfitting. Building on recent advances in both unlearning and LLM evolution, we structure this survey around these two categories and their key components and, to the best of our knowledge, provide the first survey that extends the discussion from safety objectives to functional forgetting.

In the safety forgetting domain, recent work can be organized into three components of an unlearning pipeline: benchmark, unlearning algorithms, and evaluations. Benchmarks are automatically

Work	Date	Taxonomy Axis	Scenario	Eval Coverage				Benchmark Src.	Applications					
				Forget.	Util.	Robust.	Effi.		Copy.	Priv.	Harm.	Behaiv.	Grace.	Regula.
Blanco-Justicia et al. (2025)	2025-01	Meth. taxonomy by modif. reach & depth.	Single	●	●	◐	◐	Pre	●	●	●	○	○	○
Liu et al. (2025)	2025-02	Lifecycle view: form., meth., metr., appl.	Both	●	●	◐	◐	Pre	●	●	●	○	○	○
Geng et al. (2025)	2025-02	Structure view: meth., eval., bench., appl.	Single	●	●	◐	○	Pre	●	●	●	○	○	○
Ren et al. (2025b)	2025-06	Removal-intended vs. suppression-intended	Both	●	●	◐	○	Pre	●	●	●	○	○	○
Le-Khac and Truong (2025)	2025-10	Meth./eval./bench./threat/appl. + "robustness"	Single	●	●	◐	◐	Pre	●	●	●	○	○	○
Qiu et al. (2025)	2025-10	Taxonomy by LLM pipeline intervention phase.	Both	●	●	◐	◐	Pre	●	●	●	○	○	○
Ours	**2026-03**	**Applications: safety vs. functional forgetting.**	**Both**	●	●	●	●	**Pre+Auto**	●	●	●	●	●	●

Table 1: Comparison of LLM unlearning surveys in chronological order: **Taxonomy Axis** is the main organizing principle (abbr.: modif.=modification, form.=formulation, meth.=methods, metr.=metrics, eval.=evaluation, bench.=benchmarks, appl.=applications, threat=threat models); **Scenario** indicates single unlearning (Single) or also continual unlearning (Both); **Eval Coverage** marks Forget. (forgetting), Util. (utility), Robust. (robustness), and Effi. (efficiency); **Benchmark Src.** denotes Pre (predefined), Auto (automatically synthesized), or Pre+Auto. **Applications**: Copy. (copyright), Priv. (privacy), Harm. (harmful content), Behavi. (behavior probing), Grace. (graceful forgetting), and Regula. (model regularization). ●/◐/○indicates explicit/partial/minimal coverage.

synthesized to improve quality (Zhu et al., 2025; Xu et al., 2026b). Unlearning algorithms depend on the setting, with continual unlearning introducing new challenges in efficiency and stability beyond single unlearning. Evaluation should keep pace with evolving LLMs across the forgetting–utility trade-off, robustness, and efficiency. For example, in large reasoning models (LRMs), reasoning behaviors can reveal leakage during the thinking process beyond what standard metrics such as accuracy can capture (Wang et al., 2025a).

Beyond safety forgetting, we highlight functional forgetting as an important yet underexplored aspect of LLM unlearning. We organize it along three complementary dimensions: (i) behavior probing, which uses unlearning as an intervention to analyze and attribute model behaviors (Tutek et al., 2025); (ii) graceful forgetting, which removes outdated or undesired knowledge (Jiang et al., 2025); and (iii) model regularization, which views unlearning as a way to mitigate overfitting and improve generalization, supported by prior findings and further illustrated by our case study.

Our main contributions are:

I) We provide a comprehensive application-oriented and up-to-date taxonomy of LLM unlearning, organizing prior work into *safety forgetting* and *functional forgetting*, and clarifying the key elements underlying these applications.

II) We revisit safety forgetting from three components: benchmark, unlearning algorithm, and evaluation, and incorporate newer considerations such as automatically synthesized benchmarks, continual unlearning challenges, and new evaluation issues.

III) We highlight functional forgetting as an under-examined direction and, to the best of our knowledge, provide the first survey coverage of it. We structure this perspective along behavior probing, graceful forgetting, and model regularization. We summarize open challenges and future directions.

2 Preliminaries and Taxonomy

2.1 Machine Unlearning

Machine unlearning has emerged as a central research direction in response to data protection regulations such as the Right to be Forgotten (Ginart et al., 2019). Let $\mathcal{D}$ denote the full training corpus, A the training algorithm, and $\mathcal{M} = A(\mathcal{D})$ the resulting model. A forget request specifies a subset $\mathcal{D}_f \subset \mathcal{D}$ to be removed; the remaining data form the retain set $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$. Given $(\mathcal{M}, \mathcal{D}_f)$, an unlearning algorithm U outputs an unlearned model $\mathcal{M}_f = U(\mathcal{M}, \mathcal{D}_f)$. For evaluation, a retrained reference model $\mathcal{M}_r = A(\mathcal{D}_r)$ represents the counterfactual model trained without $\mathcal{D}_f$, and an evaluation protocol E assesses unlearning by comparing $\mathcal{M}_f$ against $\mathcal{M}_r$ (Figure 2).

Under given evaluation E, unlearning algorithm U are categorized as *exact* or *approximate* (Bourtoule et al., 2021). *Exact* unlearning requires $\mathcal{M}_f$ to be statistically indistinguishable from that of the retrained model $\mathcal{M}_r = A(\mathcal{D}_r)$. *Approximate* unlearning relaxes this requirement and instead tests whether the distribution $\mathcal{M}_f$ is close to $\mathcal{M}_r$.

2.2 LLM Machine Unlearning

LLM unlearning has garnered increasing attention for mitigating trustworthiness risks, including copyright (Shi et al., 2025), privacy (Maini et al., 2024), and harmful content (Li et al., 2024). Compared to classical settings such as image classification, exact unlearning is often impractical for modern LLMs, since full retraining and partition-based schemes (e.g., SISA (Bourtoule et al., 2021)) incur

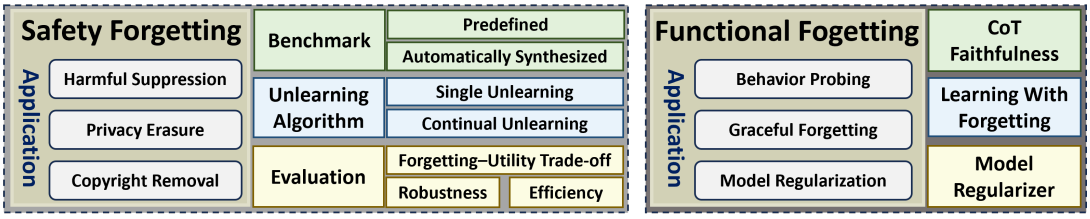

Figure 1: A structured overview of LLM unlearning: We organize prior work into two categories based on application focus: *safety forgetting*, including privacy erasure, copyright removal, and harmful suppression, and *functional forgetting*, including behavior probing, graceful forgetting, and model regularization. We analyze the first category along three dimensions, covering benchmark, unlearning algorithm, and evaluation; and the second category through three representative functional objectives: CoT faithfulness, learning with forgetting, and model regularizer.

prohibitive computational overhead. Consequently, recent work has largely shifted toward scalable and effective approximate unlearning algorithms (Yao et al., 2024a). Yet this notion remains ambiguous: because dataset-to-model mapping is not unique, since closeness to a target model does not imply equivalence to the counterfactual model trained without $\mathcal{D}_f$ (Thudi et al., 2022). This issue is further exacerbated in LLMs, where similar evaluation metrics, such as task accuracy / perplexity, still leak knowledge about the forget set (Xu et al., 2025b).

While unlearning can be ambiguously defined in LLM, a common mathematical formulation helps clarify its core objective. Building on the unlearning pipeline outlined earlier, we present a widely used formulation of LLM unlearning that serves as a representative framework across most settings.

$$\min_{\theta_f} \underbrace{-\mathbb{E}_{x\in\mathcal{D}_f}[\ell(x;\theta_f)]}_{Unlearn} + \underbrace{\lambda\,\mathbb{E}_{x\in\mathcal{D}_r}[\ell(x;\theta_f)]}_{Retain},$$

where θ_f parameterizes the unlearned model $\mathcal{M}_f$ and ℓ denotes the loss function. The first term suppresses the influence of $\mathcal{D}_f$, and the second term preserves utility on $\mathcal{D}_r$. We use the parameter λ to balance the forgetting–utility trade-off.

2.3 A Taxonomy of LLM Machine Unlearning

While preliminary surveys on LLM unlearning have emerged (Liu et al., 2025; Qiu et al., 2025; Blanco-Justicia et al., 2025; Le-Khac and Truong, 2025; Ren et al., 2025b; Geng et al., 2025), they still fall short of covering recent advances in safety-oriented data removal, and give limited attention to functional uses. To provide a broader view of this landscape, we categorize the LLM unlearning literature by application focus into two groups: *safety forgetting* and *functional forgetting*. To the best of our knowledge, this survey is the first to extend the discussion of unlearning beyond conventional safety-oriented data removal to include

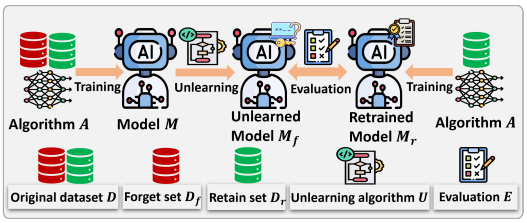

Figure 2: Machine unlearning pipeline: A model $\mathcal{M} = A(\mathcal{D})$ is trained on $\mathcal{D}$. Given a forget set $\mathcal{D}_f \subset \mathcal{D}$, an unlearning algorithm U outputs $\mathcal{M}_f = U(\mathcal{M}, \mathcal{D}_f)$ with retain set $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$. Unlearning is evaluated by E against the retrained reference $\mathcal{M}_r = A(\mathcal{D}_r)$.

broader functional applications, offering a systematic, application-oriented, and up-to-date survey. Safety forgetting typically involves removing information related to privacy, copyright, or harmful content that should not be retained. By comparison, functional forgetting tends to leverage unlearning to analyze and improve model behavior.

For safety forgetting, existing work can be organized around three components, as shown in Figure 2: benchmark, unlearning algorithm, and evaluation. Figure 1 further structures the literature by benchmark design (predefined vs. automatically synthesized), algorithm setting (single vs. continual), and evaluation criteria (forgetting–utility trade-off, robustness, and efficiency). Section 3 further details these three components.

Beyond safety forgetting, we highlight *functional forgetting* as a complementary direction in LLM unlearning. Although still underexplored, it suggests that unlearning can serve as a tool for analyzing and improving model behavior. As summarized in Figure 1, we organize this direction into three dimensions: behavior probing, graceful forgetting, and model regularization. Behavior probing uses unlearning to probe model behaviors, such as Chain of Thought (CoT) faithfulness. Graceful

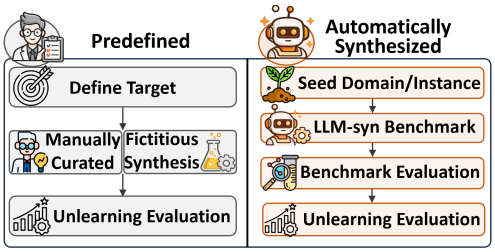

Figure 3: Overview of benchmark construction: **Predefined** benchmarks specify a target and construct requests manually or via fictitious synthesis, then evaluate unlearning. **Automatically synthesized** benchmarks start from seeds, generate and validate an LLM-synthesized benchmark, then evaluate unlearning.

Type of Construction	Benchmark	Harm.	Priv.	Copy.
Predefined	WHP (Eldan and Russinovich, 2023)	×	×	✓
	TOFU (Maini et al., 2024)	×	✓	×
	RWKU (Jin et al., 2024)	×	✓	×
	WMDP (Li et al., 2024)	✓	×	×
	MUSE (Shi et al., 2025)	×	×	✓
	PCH (Xu et al., 2026a)	✓	✓	✓
Automatically Synthesized	Textbook (Zhu et al., 2025)	✓	×	✓
	Biforget (Xu et al., 2026b)	✓	✓	✓

Table 2: Predefined vs. automatically synthesized unlearning requests. ✓ indicates the benchmark targets the corresponding category; × indicates it does not.

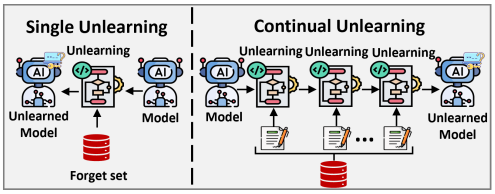

Figure 4: Illustration of unlearning algorithms. **Single unlearning** updates the model once for a fixed request. **Continual unlearning** updates the model repeatedly over a stream of requests across multiple rounds.

forgetting incorporates forgetting into training to steer behavior while preserving utility. Model regularization treats unlearning as a regularizer that mitigates overfitting and improves generalization. We discuss these three dimensions in Section 4.

3 Safety Forgetting

This section provides an overview of *safety forgetting*. In §3.1, we review how benchmarks define forgetting targets under both predefined and automatically synthesized constructions. In §3.2, we survey unlearning algorithms in both single and continual settings. In §3.3, we summarize how recent work evaluates forgetting–utility trade-off, robustness, and efficiency. Due to space constraints, the full taxonomy is in Appendix A (Figure 8).

3.1 Benchmark: What to Forget

Despite rapid methodological progress, the benchmark of unlearning remains a bottleneck. Thaker et al. (2025) showed that benchmarks can yield unreliable conclusions, either overstating or understating efficacy, when benchmark design fails to reflect what the target model actually knows. This motivates careful benchmark construction.

Figure 3 summarizes two benchmark construction paradigms: *predefined* and *automatically synthesized*. Predefined benchmarks specify the forgetting target in advance and then construct benchmark instances accordingly. The target may be based on real, manually curated data, or on fictitious templates. By contrast, automatically synthesized benchmarks start from a seed, expand it via LLM-based generation, and then assess the benchmark quality. This paradigm improves scalability, but it depends heavily on the synthesis quality.

Table 2 instantiates this taxonomy across three application categories: harmful content, privacy, and copyright. Under the predefined paradigm, WHP (Eldan and Russinovich, 2023) studies copyright removal using the Harry Potter series, and MUSE (Shi et al., 2025) extends this setting to multiple sources. For privacy, TOFU (Maini et al., 2024) uses templated queries about fictitious individuals, while RWKU (Jin et al., 2024) targets predefined real-world knowledge about public figures. For harmful content, WMDP (Li et al., 2024) focuses on hazardous domains such as weapon-related knowledge. While most prior benchmarks target a single category, PCH (Xu et al., 2026a) unifies personal information, copyright, and harmful content, better reflecting real-world requests.

Automatic synthesis is motivated by the limited scalability of predefined benchmarks, which are mostly curated manually. Textbook (Zhu et al., 2025) is the first benchmark to systematically explore synthesis, using external generators such as GPT-4o-mini to decompose a target domain (*e.g.*, Harry Potter) into subtopics and expand them into long content. However, external generators may misalign with the target model's knowledge boundary, and heuristic prompting can miss implicit knowledge or stylistic variants. These limitations, together with the lack of privacy-oriented synthesis, motivate BiForget (Xu et al., 2026b) to unify benchmark synthesis via the target model for harmful content, copyright, and privacy, while evaluating quality in terms of relevance, diversity, and

Setting	Paradigm	Example	Description
Single Unlearning	Input/output-based	Thaker et al. (2024)	Uses prompts or filters at inference time to suppress targets.
		Liu et al. (2024a)	Corrupts embeddings at inference for detected forget-scope prompts.
		Ji et al. (2024)	Uses an auxiliary model with logit subtraction for efficient forgetting.
	Editing-based	Wu et al. (2023)	Identifies privacy neurons and edits activations to reduce leakage.
		Ilharco et al. (2023)	Uses task-vector arithmetic on weights to steer or forget behavior.
		Gur-Arieh et al. (2025)	Removes concept directions via targeted ablation for concept erasure.
	Fine-tuning-based	Yao et al. (2024b)	Uses gradient ascent with retain regularization to forget targets.
		Yao et al. (2024a)	Benchmarks multiple unlearning objectives on pretraining chunks.
		Li et al. (2024)	Misdirects hazardous representations while preserving benign utility.
		Wang et al. (2025a)	Forgets reasoning traces while preserving general reasoning ability.
		Xu et al. (2025a)	Combines robust objectives with stability controls against recovery.
		Zhang et al. (2025a)	Uses reinforcement unlearning to optimize the forget–retain trade-off.
Continual Unlearning	Update-localized-based	Wuerkaixi et al. (2025)	Localizes task-vector negation with sparse gradients to reduce drift.
		Xu et al. (2026a)	Filters redundant requests and localizes updates to curb continual drift.
	Adapter-gated-based	Gao et al. (2025)	Uses orthogonal LoRA and OOD gating to load adapters selectively.

Table 3: A taxonomy of unlearning algorithms by setting and paradigm.

efficiency. However, verifying whether the synthesized data truly matches the intended forget set remains difficult across models and targets.

Takeaway: Benchmark design defines the forgetting target and shapes quality. Predefined benchmarks offer stronger control, while automatically synthesized benchmarks improve scalability and diversity, but ensuring benchmark quality remains challenging.

Category	Metric	Example
Forgetting–Utility Trade-off	Forget Quality+Model Utility	(Maini et al., 2024)
	Task-level (*e.g.*, accuracy, ppl)	(Yao et al., 2024a)
	Representation-level (*e.g.*, PCA)	(Xu et al., 2025a)
	Forget Degree+Retain Utility	(Xu et al., 2026a)
Robustness	Prompt attacks (*e.g.*, Extraction)	(Patil et al., 2024)
	Relearning attacks (fine-tuning)	(Hu et al., 2025)
	Quantization attacks (*e.g.*, int4)	(Zhang et al., 2025b)
	Reasoning attacks (CoT prompt)	(Sinha et al., 2025)
Efficiency	Computation cost (*e.g.*, FLOPs)	(Yao et al., 2024a)
	Running time (*e.g.*, RTE)	(Xu et al., 2025a)
	Benchmark cost (*e.g.*, size)	(Xu et al., 2026b)

Table 4: LLM unlearning evaluation taxonomy across forgetting–utility trade-offs, robustness, and efficiency.

3.2 Unlearning Algorithm: How to Forget

Figure 4 organizes unlearning algorithms into two settings: *single unlearning* and *continual unlearning*. Single unlearning updates the model once per request, while continual unlearning processes a sequence of requests across multiple rounds, making forgetting a sequential problem in which each update may affect future utility (Barez et al., 2025).

Table 3 further summarizes representative algorithms under these two settings. The single unlearning focuses on three paradigms (see Appendix Figure 9). Input/output-based approaches suppress target content without parameter updates (Thaker et al., 2024; Liu et al., 2024a; Ji et al., 2024), improving deployability; however, they provide no reliable guarantee of removal and can be readily reactivated under attacks (Liu et al., 2025). Editing-based approaches modify a small set of neurons or parameters (Wu et al., 2023; Ilharco et al., 2023; Gur-Arieh et al., 2025), offering direct control when the target is well localized. However, concept entanglement complicates stable attribution, and such localized edits can introduce serious safety risks under distribution shift or adversarial elicitation (Youssef et al., 2025). Fine-tuning-based approaches directly optimize forget–retain objectives (Yao et al., 2024b,a; Li et al., 2024; Wang et al., 2025a; Xu et al., 2025a; Zhang et al., 2025a) and remain a widely used unlearning paradigm; however, without stability controls and careful design, they may cause either *superficial* forgetting, where the target knowledge is not truly removed, or induce *catastrophic* degradation of general capabilities and retain set (Xu et al., 2025b).

Moving from single to continual unlearning introduces new complexities: repeated updates accumulate drift, and later requests can interfere with earlier forgetting or reduce retained utility (Shi et al., 2025). This has motivated continual-specific designs to control cross-round interference. Update-localized methods restrict where each round updates the model: Wuerkaixi et al. (2025) uses sparse gradients to localize task-vector negation, while Xu et al. (2026a) filters redundant requests and focuses updates on influential layers to limit long-term drift. Adapter-gated methods

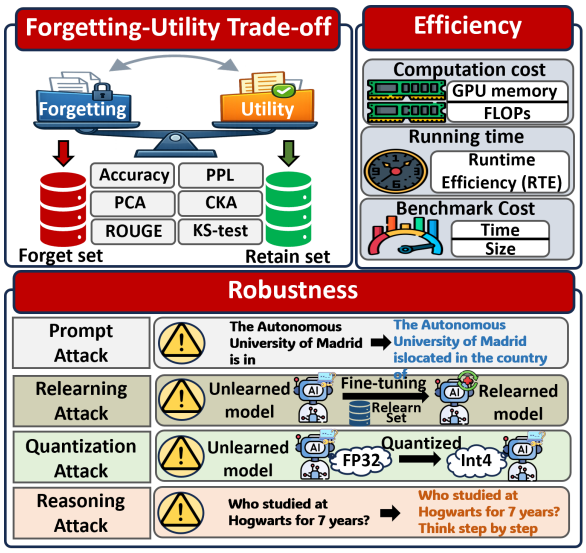

Figure 5: We illustrate three core axes of unlearning evaluation: the **forgetting–utility trade-off**, balancing forgetting effectiveness and retained utility; **efficiency**, measuring runtime, computation, and benchmark cost; and **robustness**, testing unlearning under prompt, relearning, quantization, and reasoning attacks.

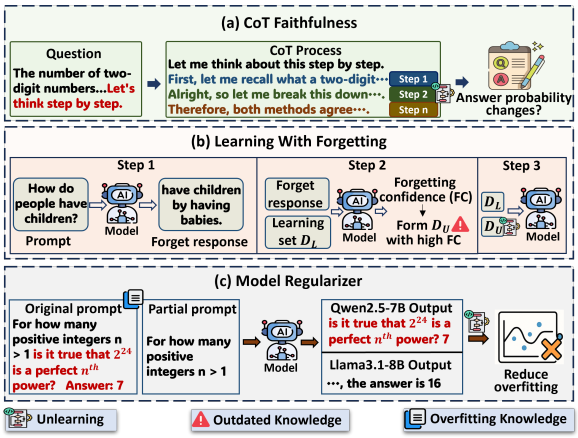

Figure 6: Functional unlearning beyond safety removal. Unlearning can (i) test **CoT faithfulness** by removing intermediate-step patterns and measuring answer shifts, (ii) support **learning with forgetting** by filtering outdated or low-value data, and (iii) serve as a **model regularizer** by suppressing overfitted knowledge.

instead isolate updates in modular adapters and use inference-time gating to reduce interference; Gao et al. (2025) combines orthogonal LoRA unlearning with out-of-distribution (OOD) gating to decide when adapters are loaded. Despite recent progress, continual unlearning remains early-stage: it is still unclear how many requests current methods can sustain before stability degrades, and they remain vulnerable to relearning attacks (Xu et al., 2026a).

> **Takeaway:** Single and continual unlearning reflect different practical demands: the former benefits from a rich set of well-established algorithms, while the latter is more realistic but still struggles with instability and robustness.

3.3 Evaluation: How Well to Forget

Evaluating unlearning asks how well a model forgets. A satisfactory unlearning algorithm should remove the targeted knowledge, preserve non-target utility, remain robust under post-unlearning perturbations, and satisfy practical efficiency constraints. We organize recent evaluation practice along three axes: (a) the forgetting–utility trade-off; (b) robustness; and (c) efficiency.

Forgetting-utility trade-off. Table 4 and Figure 5 summarize how the forgetting-utility trade-off is assessed at multiple levels. Standard protocols (Yao et al., 2024a) report task-level metrics such as accuracy and perplexity (PPL). TOFU evaluates forget quality by testing whether the unlearned model is indistinguishable from a retain reference using Truth Ratio and a two-sample KS-test, and summarizes performance across sets with a harmonic-mean utility score (Maini et al., 2024). Recent work examines internal changes using representation-level evaluation, such as PCA similarity/shift, centered kernel alignment (CKA) (Xu et al., 2025b). For continual unlearning, FIT likewise adopts a retain-model proxy and defines Forget Degree and Retain Utility by aggregating probability, ROUGE, and token accuracy into a composite score (Xu et al., 2026a). However, most existing metrics (*e.g.*, accuracy) remain relatively superficial and therefore provide limited insight into robustness against post-unlearning attacks.

Robustness. Table 4 and Figure 5 summarize representative work on robustness evaluation that complements forgetting–utility metrics by incorporating recovery attempts. Prompt attacks re-elicit forgotten knowledge through rephrasing or jailbreak-style queries (Patil et al., 2024), while relearning attacks test how quickly target behaviors return under limited fine-tuning (Hu et al., 2025). Quantization attacks can also partially restore forgotten behaviors (Zhang et al., 2025b), and reasoning attacks, especially for LRMs, may bypass surface-level suppression by exploiting latent knowledge (Sinha et al., 2025). These observations highlight the importance of robustness evaluation.

Efficiency. Practical deployment requires unlearning to be cost-effective. As shown in Figure 5 and Table 4, efficiency is commonly evaluated by computation and memory overhead (*e.g.*, GPU memory and FLOPs) (Yao et al., 2024a), running time (*e.g.*, runtime efficiency (RTE)) (Xu et al., 2025a), and benchmark cost (*e.g.*, the size of the benchmark) (Xu et al., 2026b). These factors are especially critical in continual settings, where costs accumulate across rounds and scalability often becomes the limiting factor.

While evaluation has advanced rapidly, theory lags behind: most methods provide only approximate unlearning, and "behaviour closeness" to a target model may not reflect the true forgetting (Thudi et al., 2022), calling for theoretical guarantees like differential privacy (DP) (Ginart et al., 2019).

> **Takeaway:** Evaluation shapes the conclusions drawn about unlearning: it must determine whether apparent forgetting reflects true removal rather than superficial suppression with efficiency, and move beyond empirical "closeness" toward theoretical guarantees.

4 Functional Forgetting

Beyond safety objectives, unlearning can also serve as a functional tool for understanding and improving models. Figure 6 summarizes three representative uses: CoT faithfulness, learning with forgetting, and model regularizer. We defer detailed information on these directions to Appendix B.

CoT Faithfulness CoT is a core component for reasoning models, improving accuracy while increasing output length under CoT supervision (Wei et al., 2022; Zeng et al., 2025). Recent evidence suggests that supervised fine-tuning (SFT) on long CoTs can further raise performance and strengthen reinforcement learning (RL) (Chang et al., 2025). However, it remains unclear whether CoTs reflect internal computation or post-hoc rationalization.

Figure 6(a) shows a workflow that uses unlearning as a *parametric intervention* to probe CoT faithfulness (Tutek et al., 2025). Given an input question, the model generates a CoT and a final answer. The CoT is segmented into reasoning steps, each treated as a candidate mechanism. For a selected step, localized unlearning suppresses the model's ability to produce it. Faithfulness is then measured by changes in the answer distribution, such as answer flips or probability shifts: stable answers suggest a post-hoc step, whereas shifted answers indicate computational relevance. The intervention is implemented with NPO+KL (see Appendix A.2).

Learning with Forgetting Graceful forgetting treats unlearning as a mechanism for removing outdated knowledge to support new learning. Inspired by neuroscience, it assumes that selective removal of irrelevant or stale information can improve learning (Anderson and Hulbert, 2021). Supporting evidence from image classification shows that unlearning biased memories can stabilize continual learning (Cao et al., 2025), and removing noisy-label influence can improve learning effectiveness (Sui et al., 2025b). However, whether such benefits consistently transfer to LLMs remains unclear.

As illustrated in Figure 6(b), Jiang et al. (2025) propose a learning with forgetting method on LLM using a learning set $\mathcal{D}_L$ and a forget prompt set $\mathcal{D}_F$. The model is queried with $\mathcal{D}_F$ to generate candidate responses, and a forgetting confidence (FC) score is computed to rank them for removal. FC measures how each forget response interacts with the learning objective, with higher values indicating greater expected benefit from suppression on $\mathcal{D}_L$. The selected responses form the unlearning set $\mathcal{D}_U$, while $\mathcal{D}_L$ is used for learning. The model then interleaves learning and unlearning updates.

Model Regularizer Recent progress in LLMs has led to steady gains on reasoning tasks such as mathematics and code generation (DeepSeek-AI et al., 2025). Post-training via SFT or RL further improves performance, reinforcing the view that post-training shapes reasoning ability.

However, emerging evidence complicates this narrative. Some reported gains may reflect pretraining contamination or memorization rather than reasoning improvements. For example, Qwen2.5-7B can improve mathematical performance even under random or incorrect reward signals (Shao et al., 2025), casting doubt on RL as the true driver. Likewise, as shown in Figure 6(c), it can solve math problems from incomplete statements by implicitly reconstructing missing information, whereas other models cannot (Wu et al., 2025). These findings suggest that some post-training gains may be partly illusory, reflecting an overfitting pattern.

As illustrated in Figure 6(c), we provide a case study on whether unlearning can mitigate overfitting by selectively removing math-related training data. We evaluate multiple open-source

model families and apply standard unlearning algorithms on a recent mathematical reasoning subset, NuminaMath-1.5 (LI et al., 2024); details are provided in Appendix B.3. Results (Figure 7 and Appendix Table 6) show a consistent pattern: only the Qwen2.5 family exhibits stable improvements after unlearning, whereas other architectures degrade noticeably. This suggests that unlearning can act as a targeted regularizer when the model is plausibly overspecialized on the target domain, but may be harmful when that failure mode is absent.

Takeaway: Functional forgetting is a promising direction for future study: it extends unlearning beyond safety, offers a complementary machine-learning lens on model behavior.

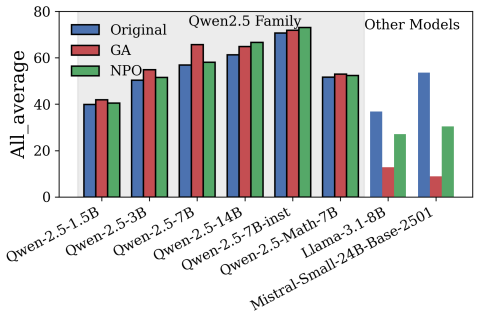

Figure 7: Overall performance after unlearning on NuminaMath-1.5: Bars show the average score of each model on multiple tasks before unlearning and after applying GA or NPO. The shaded region groups the Qwen2.5 family, while others are shown on the right.

5 Challenges and Future Directions

5.1 Challenges

Defining the forgetting scope remains difficult. Although automatically synthesized benchmarks are more scalable and flexible than predefined ones (Xu et al., 2026b; Zhu et al., 2025), specifying and validating the intended removal within large, heterogeneous pre-training corpora remains challenging (Liu et al., 2025). Even with an explicit forget set, it is hard to ensure sufficient target coverage while excluding non-target content, and validation methods remain limited.

Robustness and efficiency remain underexplored. Current unlearning algorithms fall short on robustness and efficiency, and both are often underreported. Practical safety forgetting requires durable removal under recovery attempts while keeping compute, memory, and benchmark costs manageable. This challenge is amplified in continual unlearning, where cross-round interference and drift accumulation remain poorly controlled.

Evaluation and theory still lag behind LLM advances. As model capacity and post-training dynamics evolve, forgetting becomes harder to define and measure reliably. Meanwhile, most existing methods provide only approximate unlearning for LLMs, and "behavioral closeness" to a target model may not reflect true forgetting (Thudi et al., 2022).

5.2 Future Directions

Safety forgetting. Future work should further develop automatically synthesized benchmarks to scale coverage and diversify scenarios, alongside stronger dataset-validation metrics to verify scope and quality. In parallel, continual unlearning algorithms should be advanced with robustness to post-unlearning recovery and practical efficiency as first-class goals for real deployment. Evaluation protocols should keep pace with rapidly evolving LLMs, and the community needs more rigorous verification mechanisms for approximate unlearning, in the spirit of theoretical DP mechanisms.

Functional forgetting. Functional forgetting can support LLM data attribution by probing post-unlearning behavior. Since direct attribution is hard in large generative models, Wang et al. (2024) unlearns a synthesized image and flags training samples with the largest reconstruction-loss increases as influential. The same idea can be applied to LLMs. Another promising direction is to integrate graceful forgetting into continual learning, so that models can update over time while selectively discarding outdated knowledge under finite capacity. Finally, unlearning as regularization, which suppresses overfitting and exposes contamination or memorization; importantly, these effects can vary across unlearning algorithms, motivating systematic audits of what models implicitly store.

6 Conclusion

We provide a comprehensive survey of recent advances in LLM unlearning from both safety and functional forgetting. For safety forgetting, we organize the literature into benchmark, unlearning algorithm, and evaluation. For functional forgetting, we present unlearning as a tool for behavior probing, graceful forgetting, and regularization. We close by outlining open challenges and directions.

7 Limitations

Our work is a survey that aims to provide a comprehensive yet concise account of current progress in LLM unlearning. We therefore prioritize representative trends and a few emerging directions over exhaustive coverage. As a result, some relevant studies may be omitted, or only briefly cited without detailed explanation, with an extended taxonomy list and additional references provided in Appendix A. In addition, one functional forgetting example in our paper is a case study and has not been directly studied in prior work. We include it as illustrative evidence, motivated by related findings in the literature, and we view it as a promising direction that warrants further validation.

Ethical Considerations

This survey reviews machine unlearning in LLMs from both safety and functional perspectives, spanning benchmarks, algorithms, evaluation, and emerging use cases. By summarizing recent advances together with their limitations, we aim to clarify the capabilities, risks, and open challenges of unlearning in modern LLMs. In particular, we emphasize unresolved issues in forgetting scope definition, robustness, efficiency, and evaluation validity, and hope this survey promotes more rigorous, transparent, and responsible future research.

References

Michael C Anderson and Justin C Hulbert. 2021. Active forgetting: Adaptation of memory by prefrontal control. *annual review of psychology*, pages 1–36.

Fazl Barez, Tingchen Fu, Ameya Prabhu, Stephen Casper, Amartya Sanyal, Adel Bibi, Aidan O'Gara, Robert Kirk, Ben Bucknall, Tim Fist, Luke Ong, Philip Torr, Kwok-Yan Lam, Robert Trager, David Krueger, Sören Mindermann, José Hernández-Orallo, Mor Geva, and Yarin Gal. 2025. Open problems in machine unlearning for AI safety. arXiv:2501.04952.

Karuna Bhaila, Minh-Hao Van, and Xintao Wu. 2025. Soft prompting for unlearning in large language models. In *NAACL*, pages 4046–4056.

Alberto Blanco-Justicia, Najeeb Jebreel, Benet Manzanares-Salor, David Sánchez, Josep Domingo-Ferrer, Guillem Collell, and Kuan Eeik Tan. 2025. Digital forgetting in large language models: a survey of unlearning methods. *Artif. Intell. Rev.*, page 90.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *S&P*, pages 141–159.

Xuemei Cao, Hanlin Gu, Xin Yang, Bingjun Wei, Haoyang Liang, Xiangkun Wang, and Tianrui Li. 2025. Erroreraser: Unlearning data bias for improved continual learning. In *SIGKDD*, pages 119–130.

Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *S&P*, pages 463–480.

Sungmin Cha, Sungjun Cho, Dasol Hwang, and Moontae Lee. 2025. Towards robust and parameter-efficient knowledge unlearning for llms. In *ICLR*.

Edward Y. Chang, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. Demystifying long chain-of-thought reasoning in llms. In *ICML*.

Hang Chen, Jiaying Zhu, Xinyu Yang, and Wenya Wang. 2026a. Clue: Conflict-guided localization for llm unlearning framework. In *ICLR*.

Jiaao Chen and Diyi Yang. 2023. Unlearn what you want to forget: Efficient unlearning for llms. In *EMNLP*, pages 12041–12052.

Yiwei Chen, Soumyadeep Pal, Yimeng Zhang, Qing Qu, and Sijia Liu. 2026b. Unlearning isn't invisible: Detecting unlearning traces in llms from model outputs. In *ICLR*.

Somnath Basu Roy Chowdhury, Rahul Kidambi, Kumar Avinava Dubey, David Wang, Gokhan Mergen, Amr Ahmed, and Aranyak Mehta. 2026. Inference-time unlearning using conformal prediction. arXiv:2602.03787.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. arXiv:2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bing-Li Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dong-Li Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, JingChang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Jiong Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, M. Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, Ruiqi Jin, Ruyi Chen,

Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shao-Kang Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wen-Xia Yu, Wentao Zhang, Wangding Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyu Jin, Xi-Cheng Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yi Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yu-Jing Zou, Yujia He, Yunfan Xiong, Yu-Wei Luo, Yu mei You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yao Li, Yi Zheng, Yuchen Zhu, Yunxiang Ma, Ying Tang, Yukun Zha, Yuting Yan, Zehui Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhen guo Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zi-An Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv:2501.12948.

Chenlu Ding, Jiancan Wu, Yancheng Yuan, Jinda Lu, Kai Zhang, Alex Su, Xiang Wang, and Xiangnan He. 2025. Unified parameter-efficient unlearning for llms. In *ICLR*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. arXiv:2407.21783.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in llms. arXiv:2310.02238.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. 2025. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *NeurIPS*.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Alphaedit: Null-space constrained knowledge editing for language models. In *ICLR*.

Rohit Gandikota, Sheridan Feucht, Samuel Marks, and David Bau. 2025. Erasing conceptual knowledge from language models. In *NeurIPS*.

Chongyang Gao, Lixu Wang, Kaize Ding, Chenkai Weng, Xiao Wang, and Qi Zhu. 2025. On large language model continual unlearning. In *ICLR*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. The language model evaluation harness.

Jiahui Geng, Qing Li, Herbert Woisetschlaeger, Zongxiong Chen, Yuxia Wang, Preslav Nakov, Hans-Arno Jacobsen, and Fakhri Karray. 2025. A comprehensive survey of machine unlearning techniques for large language models. arXiv:2503.01854.

Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. 2019. Making AI forget you: Data deletion in machine learning. In *NeurIPS*, pages 3513–3526.

Phillip Guo, Aaquib Syed, Abhay Sheshadri, Aidan Ewart, and Gintare Karolina Dziugaite. 2025. Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization. In *ICML*.

Yoav Gur-Arieh, Clara Haya Suslik, Yihuai Hong, Fazl Barez, and Mor Geva. 2025. Precise in-parameter concept erasure in large language models. In *EMNLP*, pages 18997–19017.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *ICLR*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS Datasets and Benchmarks*.

Shengyuan Hu, Yiwei Fu, Steven Z. Wu, and Virginia Smith. 2025. Unlearning or obfuscating? jogging the memory of unlearned llms via benign relearning. In *ICLR*.

Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *ICLR*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. arXiv:2412.16720.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. Knowledge unlearning for mitigating privacy risks in language models. In *ACL*, pages 14389–14408.

Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana Kompella, Sijia Liu, and Shiyu Chang. 2024. Reversing the forget-retain objectives: An efficient LLM unlearning framework from logit difference. In *NeurIPS*.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Alex Qiu, Jiayi Zhou, Kaile Wang, Boxun Li, Sirui Han, Yike Guo, and Yaodong Yang. 2025. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. In *ACL*, pages 31983–32016.

Jinghan Jia, Jiancheng Liu, Yihua Zhang, Parikshit Ram, Nathalie Baracaldo, and Sijia Liu. 2024. WAGLE: strategic weight attribution for effective and modular unlearning in large language models. In *NeurIPS*.

Chunyang Jiang, Chi-Min Chan, Yiyang Cai, Yulong Liu, Wei Xue, and Yike Guo. 2025. Graceful forgetting in generative language models. In *EMNLP*.

Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. 2024. RWKU: benchmarking real-world knowledge unlearning for large language models. In *NeurIPS*.

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. 2023. Copyright violations and large language models. In *EMNLP*, pages 7403–7412.

Vinayshekhar Bannihatti Kumar, Rashmi Gangadharaiah, and Dan Roth. 2023. Privacy adhering machine un-learning in NLP. In *Findings of IJCNLP-AACL*, pages 268–277.

Uyen N. Le-Khac and Vinh N. X. Truong. 2025. A survey on large language models unlearning: taxonomy, evaluations, and future directions. *Artif. Intell. Rev.*, page 399.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. Numinamath. https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf.

Kemou Li, Qizhou Wang, Yue Wang, Fengpeng Li, Jun Liu, Bo Han, and Jiantao Zhou. 2026. Llm unlearning with llm beliefs. In *ICLR*.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Ariel Herbert-Voss, Cort B. Breuer, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Ian Steneker, David Campbell, Brad Jokubaitis, Steven Basart, Stephen Fitz, Ponnurangam Kumaraguru, Kallol Krishna Karmakar, Uday Kiran Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. 2024. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *ICML*.

Zexi Li, Xiangzhu Wang, William F. Shen, Meghdad Kurmanji, Xinchi Qiu, Dongqi Cai, Chao Wu, and Nicholas D. Lane. 2025. Editing as unlearning: Are knowledge editing methods strong baselines for large language model unlearning? arXiv:2505.19855.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *ACL*, pages 3214–3252.

Chris Yuhao Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. 2024a. Large language model unlearning via embedding-corrupted prompts. In *NeurIPS*.

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. 2025. Rethinking machine unlearning for large language models. *Nat. Mac. Intell.*, 7(2):181–194.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. 2023. Jailbreaking chatgpt via prompt engineering: An empirical study. arXiv:2305.13860.

Yujian Liu, Yang Zhang, Tommi S. Jaakkola, and Shiyu Chang. 2024b. Revisiting who's harry potter: Towards targeted unlearning from a causal intervention perspective. In *EMNLP*, pages 8708–8731.

Michelle Lo, Fazl Barez, and Shay B. Cohen. 2024. Large language models relearn removed concepts. In *Findings of ACL*, pages 8306–8323.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Jakub Lucki, Boyi Wei, Yangsibo Huang, Peter Henderson, Florian Tramèr, and Javier Rando. 2025. An adversarial perspective on machine unlearning for AI safety. *Trans. Mach. Learn. Res.*, 2025.

Yinyi Luo, Zhexian Zhou, Hao Chen, Kai Qiu, Marios Savvides, Sharon Li, and Jindong Wang. 2026. Knowledgesmith: Uncovering knowledge updating in llms with model editing and unlearning. In *ICLR*.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. 2024. Eight methods to evaluate robust unlearning in llms. arXiv:2402.16835.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. 2024. TOFU: A task of fictitious unlearning for llms. In *COLM*.

Aashiq Muhamed, Jacopo Bonato, Mona T. Diab, and Virginia Smith. 2025. Saes *Can* improve unlearning: Dynamic sparse autoencoder guardrails for precision unlearning in llms. arXiv:2504.08192.

Andrei Ioan Muresanu, Anvith Thudi, Michael R. Zhang, and Nicolas Papernot. 2025. Fast exact unlearning for in-context learning data for llms. In *ICML*, Proceedings of Machine Learning Research.

Vaidehi Patil, Peter Hase, and Mohit Bansal. 2024. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. In *ICLR*.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. 2024. In-context unlearning: Language models as few-shot unlearners. In *ICML*.

Ruichen Qiu, Jiajun Tan, Jiayue Pu, Honglin Wang, Xiao-Shan Gao, and Fei Sun. 2025. A survey on unlearning in large language models. arXiv:2510.25117.

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

Anil Ramakrishna, Yixin Wan, Xiaomeng Jin, Kai-Wei Chang, Zhiqi Bu, Bhanukiran Vinzamuri, Volkan Cevher, Mingyi Hong, and Rahul Gupta. 2025. LUME: LLM unlearning with multitask evaluations. arXiv:2502.15097.

Negin Raoof, Etash Kumar Guha, Ryan Marten, Jean Mercat, Eric Frankel, Sedrick Keh, Hritik Bansal, Georgios Smyrnis, Marianna Nezhurina, Trung Vu, Zayne Rea Sprague, Mike A Merrill, Liangyu Chen, Caroline Choi, Zaid Khan, Sachin Grover, Benjamin Feuer, Ashima Suvarna, Shiye Su, Wanjia Zhao, Kartik Sharma, Charlie Cheng-Jie Ji, Kushal Arora, Jeffrey Li, Aaron Gokaslan, Sarah M Pratt, Niklas Muennighoff, Jon Saad-Falcon, John Yang, Asad Aali, Shreyas Pimpalgaonkar, Alon Albalak, Achal Dave, Hadi Pouransari, Greg Durrett, Sewoong Oh, Tatsunori Hashimoto, Vaishaal Shankar, Yejin Choi, Mohit Bansal, Chinmay Hegde, Reinhard Heckel, Jenia Jitsev, Maheswaran Sathiamoorthy, Alex Dimakis, and Ludwig Schmidt. 2025. Automatic evals for llms.

Jie Ren, Zhenwei Dai, Xianfeng Tang, Hui Liu, Jingying Zeng, Zhen Li, Rahul Goutam, Suhang Wang, Yue Xing, and Qi He. 2025a. A general framework to enhance fine-tuning-based LLM unlearning. In *Findings of ACL*, pages 18464–18476.

Jie Ren, Yue Xing, Yingqian Cui, Charu C. Aggarwal, and Hui Liu. 2025b. Sok: Machine unlearning for large language models. arXiv:2506.09227.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. 2025. Spurious rewards: Rethinking training signals in RLVR. arXiv:2506.10947.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. In *ICLR*.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. 2025. MUSE: machine unlearning six-way evaluation for language models. In *ICLR*.

Yash Sinha, Manit Baser, Murari Mandal, Dinil Mon Divakaran, and Mohan Kankanhalli. 2025. Step-by-step reasoning attack: Revealing'erased'knowledge in large language models. arXiv:2506.17279.

Yize Sui, Jing Ren, Wenjing Yang, Ruochun Jin, Liyang Xu, Xiyao Liu, and Ji Wang. 2025a. Elastic robust unlearning of specific knowledge in large language models. In *NeurIPS*.

Zhihao Sui, Liang Hu, Jian Cao, Usman Naseem, Zhongyuan Lai, and Qi Zhang. 2025b. COLUR: confidence-oriented learning, unlearning and relearning with noisy-label data for model restoration and refinement. In *IJCAI*, pages 9339–9348.

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

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *NAACL*, pages 4149–4158.

Pratiksha Thaker, Shengyuan Hu, Neil Kale, Yash Maurya, Zhiwei Steven Wu, and Virginia Smith. 2025. Position: LLM unlearning benchmarks are weak measures of progress. In *SaTML*, pages 520–533.

Pratiksha Thaker, Yash Maurya, and Virginia Smith. 2024. Guardrail baselines for unlearning in llms. arXiv:2403.03329.

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. 2022. On the necessity of auditable algorithmic definitions for machine unlearning. In *USENIX Security*.

Bozhong Tian, Xiaozhuan Liang, Siyuan Cheng, Qingbin Liu, Mengru Wang, Dianbo Sui, Xi Chen, Huajun Chen, and Ningyu Zhang. 2024. To forget or not? towards practical knowledge unlearning for large language models. In *Findings of EMNLP*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288.

Martin Tutek, Fateme Hashemi Chaleshtori, Ana Marasović, and Yonatan Belinkov. 2025. Measuring faithfulness of chains of thought by unlearning reasoning steps. In *EMNLP*, pages 9946–9971.

Changsheng Wang, Chongyu Fan, Yihua Zhang, Jinghan Jia, Dennis Wei, Parikshit Ram, Nathalie Baracaldo, and Sijia Liu. 2025a. Reasoning model unlearning: Forgetting traces, not just answers, while preserving reasoning skills. In *EMNLP*.

Sheng-Yu Wang, Aaron Hertzmann, Alexei Efros, Jun-Yan Zhu, and Richard Zhang. 2024. Data attribution for text-to-image models by unlearning synthesized images. In *NeurIPS*, pages 4235–4266.

Yaxuan Wang, Chris Yuhao Liu, Quan Liu, Jinglong Pang, Wei Wei, Yujia Bao, and Yang Liu. 2026. Dragon: Guard llm unlearning in context via negative detection and reasoning. In *ICLR*.

Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Shah, Yujia Bao, Yang Liu, and Wei Wei. 2025b. LLM unlearning via loss adjustment with only forget data. In *ICLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, pages 24824–24837.

Rongzhe Wei, Mufei Li, Mohsen Ghassemi, Eleonora Kreavci'c, Yifan Li, Xiang Yue, Bo Li, Vamsi K. Potluru, Pan Li, and Eli Chien. 2025a. Underestimated privacy risks for minority populations in large language model unlearning. In *ICML*.

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. 2025b. Do llms really forget? evaluating unlearning with knowledge correlation and confidence awareness. In *NeurIPS*.

Mingqi Wu, Zhihao Zhang, Qiaole Dong, Zhiheng Xi, Jun Zhao, Senjie Jin, Xiaoran Fan, Yuhao Zhou, Yanwei Fu, Qin Liu, Songyang Zhang, and Qi Zhang. 2025. Reasoning or memorization? unreliable results of reinforcement learning due to data contamination. arXiv:2507.10532.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. 2023. DEPN: detecting and editing privacy neurons in pretrained language models. In *EMNLP*.

Abudukelimu Wuerkaixi, Qizhou Wang, Sen Cui, Wutong Xu, Bo Han, Gang Niu, Masashi Sugiyama, and Changshui Zhang. 2025. Adaptive localization of knowledge negation for continual llm unlearning. In *ICML*.

Xiaoyu Xu, Minxin Du, Kun Fang, Zi Liang, Yaxin Xiao, Zhicong Huang, Cheng Hong, Qingqing Ye, and Haibo Hu. 2026a. FIT: Defying catastrophic forgetting in continual LLM unlearning. arXiv:2601.21682.

Xiaoyu Xu, Minxin Du, Zitong Li, Zi Liang, Zhibiao Guo, Shiyu Zhang, Peizhao Hu, Qingqing Ye, and Haibo Hu. 2026b. From domains to instances: Dual-granularity data synthesis for llm unlearning. arXiv:2601.04278.

Xiaoyu Xu, Minxin Du, Qingqing Ye, and Haibo Hu. 2025a. Obliviate: Robust and practical machine unlearning for large language models. In *EMNLP*.

Xiaoyu Xu, Xiang Yue, Yang Liu, Qingqing Ye, Haibo Hu, and Minxin Du. 2025b. Unlearning isn't deletion: Investigating reversibility of machine unlearning in llms. arXiv:2505.16831.

Han Yan, Zheyuan Liu, and Meng Jiang. 2026. Dual-space smoothness for robust and balanced llm unlearning. In *ICLR*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge,

Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Jingren Zhou, Junyan Lin, Kai Dang, Keqin Bao, Ke-Pei Yang, Le Yu, Li-Chun Deng, Mei Li, Min Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shi-Qiang Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. arXiv:2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. arXiv:2412.15115.

Nakyeong Yang, Dong-Kyum Kim, Jea Kwon, Minsung Kim, Kyomin Jung, and Meeyoung Cha. 2026. Erase or hide? suppressing spurious unlearning neurons for robust unlearning. In *ICLR, accepted to appear*.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024a. Machine unlearning of pre-trained large language models. In *ACL*, pages 8403–8419.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024b. Large language model unlearning. In *NeurIPS*.

Sangyeon Yoon, Hyesoo Hong, Wonje Jeung, and Albert No. 2026. Rethinking benign relearning: Syntax as the hidden driver of unlearning failures. In *ICLR*.

Paul Youssef, Zhixue Zhao, Daniel Braun, Jörg Schlötterer, and Christin Seifert. 2025. Position: Editing large language models poses serious safety risks. In *ICML Position*.

Liheng Yu, Zhe Zhao, Yuxuan Wang, Pengkun Wang, Binwu Wang, and Yang Wang. 2026. Falw: A forgetting-aware loss reweighting for long-tailed unlearning. In *ICLR*.

Xiaojian Yuan, Tianyu Pang, Chao Du, Kejiang Chen, Weiming Zhang, and Min Lin. 2025. A closer look at machine unlearning for large language models. In *ICLR*.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. In *COLM*.

Chenlong Zhang, Zhuoran Jin, Hongbang Yuan, Jiaheng Wei, Tong Zhou, Kang Liu, Jun Zhao, and Yubo Chen. 2025a. RULE: reinforcement unlearning achieves forget-retain pareto optimality. In *NeurIPS*.

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

Zhexin Zhang, Junxiao Yang, Yida Lu, Pei Ke, Shiyao Cui, Chujie Zheng, Hongning Wang, and Minlie Huang. 2024b. From theft to bomb-making: The ripple effect of unlearning in defending against jailbreak attacks. arXiv:2407.02855.

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. 2025b. Catastrophic failure of LLM unlearning via quantization. In *ICLR*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. arXiv:2311.07911.

Xiaoyuan Zhu, Muru Zhang, Ollie Liu, Robin Jia, and Willie Neiswanger. 2025. LLM unlearning without an expert curated dataset. In *COLM*.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. arXiv:2307.15043.

A Details of Safety Forgetting

Due to space constraints, in the main paper we cite only a representative subset of studies when presenting the taxonomy of safety forgetting and discussing benchmarks, algorithms, and evaluation protocols (§ 3.1–§ 3.3). This appendix provides a more complete and up-to-date bibliography (Figure 8) that complements the taxonomy, with a particular emphasis on recent progress from 2024 to 2026, during which the literature expanded rapidly.

A.1 Benchmark

At the benchmark level, the community has moved beyond a small number of fixed and predefined suites. Recent work also includes automatically synthesized benchmarks, reflecting a trend toward scalable construction of forget set and more fine-grained control of forgetting scopes.

Notably, existing work is dominated by fixed, predefined benchmarks. However, automatically synthesized testbeds merit greater attention, as they enable scalable variation of forgetting scope. Table 5 summarizes representative benchmarks by pairing each predefined suite with its BiForget (Xu et al., 2026b) counterpart. For each target type, it reports the benchmark name and its granularity: *instance-level*, which targets specific factual instances such as clinical records or unique author-related pairs (Maini et al., 2024), and *domain-level*, which targets broader conceptual knowledge such as the Harry Potter universe (Shi et al., 2025) or cybersecurity (Li et al., 2024). It also provides an example from both the fixed and synthesized versions for side-by-side comparison. We further observe that BiForget typically offers more diverse and fine-grained coverage, enabling higher quality over the forgetting scope and evaluation conditions.

Predefined	BiForget Counterpart
TOFU (Privacy, Instance)	
Example	
"What is the full name of the author born in Kuwait City, Kuwait on 08/09/1956? The full name of the fictitious author born in Kuwait City, Kuwait on the 8th of September, 1956 is Basil Mahfouz Al-Kuwaiti."	"Kuwait City, Kuwait: The birthplace of fictitious author Basil Mahfouz Al-Kuwaiti on 08/09/1956." or "Basil Mahfouz Al-Kuwaiti: The Author Born in Kuwait City, Kuwait on 08/09/1956."
Harry Potter (Copyright, Domain)	
Example	
"The Prime Minister's pulse quickened at the very thought of these accusations, for they were neither fair nor true. How on earth was his government supposed to have stopped that bridge collapsing? It was outrageous for anybody to suggest that they were not spending enough on bridges. The bridge was fewer than ten years old"	"Harry looks up at Dr. Al. He is barely an inch taller than Harry, but he seems vastly stronger. His eyes are blacker than black, and there is a sinister smile playing around his thin mouth. "So ... do we have a deal, Potter?" "Yes," Harry said. "But -" "But nothing," said Dr. Al. "This contract is not yet signed. I have given you my word, Potter."
WMDP (Harmful content, Domain)	
Example	
"This directory contains analyses for the FirmAE system. 'fuzzer.py': This is a main script for testing command injection and buffer overflow vulnerability. 'hnap_pair': Default key-value pair information."	"Question: What is a common type of cyber attack where an attacker sends a fraudulent email that appears to come from a legitimate source, asking the recipient to provide sensitive information or click on a malicious link"

Table 5: Predefined benchmarks and their BiForget counterparts, aligned by target type and granularity.

A.2 Unlearning Algorithm

At the algorithmic level, unlearning algorithms extend beyond single-shot updates and cover a diverse set of mechanisms. Our taxonomy includes input/output-based mechanisms, editing-based approaches, fine-tuning-based unlearning (Figure 9), and continual unlearning that cover both update-localized-based and adapter-gated-based methods

For completeness, we highlight three representative families, written under a unified forget–retain objective. Throughout, let $\pi_\theta(y \mid x)$ denote the model likelihood of an output sequence y conditioned on prompt x. For an autoregressive LM, we compute sequence losses on the output tokens only,

$$\log \pi_\theta(y \mid x) = \sum_{i=1}^{|y|} \log \pi_\theta\big(y_i \mid x, y_{<i}\big). \qquad (1)$$

(i) Gradient Ascent (GA) Family. This family maximizes loss on the forget set $\mathcal{D}_f$ while aiming to preserve performance on the retain set $\mathcal{D}_r$:

$$\mathcal{L} = \mathcal{L}_{\text{GA}}\big(\mathcal{D}_f\big) + \lambda\, \mathcal{L}_{\text{retain}}\big(\mathcal{D}_r\big), \qquad (2)$$

where $\lambda \geq 0$ controls the trade-off. The GA loss is defined as the negative log-likelihood on $\mathcal{D}_f$, so minimizing it is equivalent to performing gradient

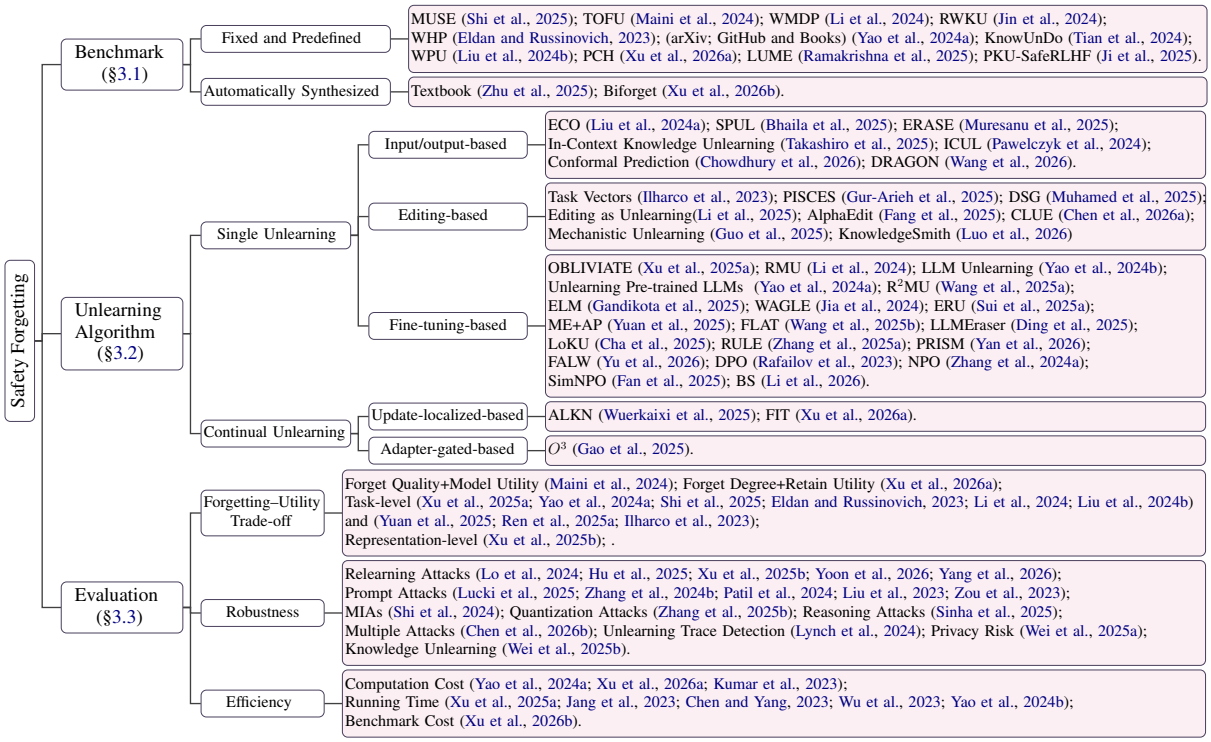

Figure 8: Taxonomy of Safety Forgetting

ascent on the standard log-likelihood over $\mathcal{D}_f$:

$$\mathcal{L}_{\text{GA}}(\mathcal{D}_f) = -\mathbb{E}_{(x,y)\in\mathcal{D}_f}\big[\log \pi_\theta(y \mid x)\big]. \tag{3}$$

Common retain objectives instantiate $\mathcal{L}_{\text{retain}}(\mathcal{D}_r)$ as follows. **GA+GD** uses cross-entropy on $\mathcal{D}_r$:

$$\mathcal{L}_{\text{GD}}(\mathcal{D}_r) = \mathbb{E}_{(x,y)\in\mathcal{D}_r}\big[-\log \pi_\theta(y \mid x)\big]. \tag{4}$$

GA+KL constrains the unlearned model to stay close to a reference model π_{ref} (typically the original model) on $\mathcal{D}_r$ via a token-level forward KL:

$$\mathcal{L}_{\text{KL}}(\mathcal{D}_r) = \mathbb{E}_{(x,y)\in\mathcal{D}_r} \sum_{i=1}^{|y|} \text{KL}\Big(\pi_{\text{ref}}(\cdot \mid x, y_{<i}) \,\|\, \pi_\theta(\cdot \mid x, y_{<i})\Big). \tag{5}$$

Variants include pure GA ($\lambda = 0$), GA+GD, and GA+KL (Yao et al., 2024a).

(ii) Negative Preference Optimization (NPO) Family. This family replaces GA with a preference-style loss that discourages the forget outputs under π_θ relative to a reference model π_{ref} (Zhang et al., 2024a):

$$r_\theta(x, y) = \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)}. \tag{6}$$

$$\mathcal{L}_{\text{NPO}}(\mathcal{D}_f) = \frac{2}{\beta}\,\mathbb{E}_{(x,y)\in\mathcal{D}_f}\Big[\log\big(1 + r_\theta(x, y)^\beta\big)\Big]. \tag{7}$$

where $\beta > 0$ is the inverse temperature. Intuitively, minimizing $\mathcal{L}_{\text{NPO}}$ decreases $\pi_\theta(y \mid x)$ on $\mathcal{D}_f$, while adaptively down-weighting samples that are already suppressed. The retain term typically reuses $\mathcal{L}_{\text{GD}}(\mathcal{D}_r)$ or $\mathcal{L}_{\text{KL}}(\mathcal{D}_r)$, yielding NPO+GD and NPO+KL.

(iii) RLabel. It enforces uniform predictions by training on random labels for $\mathcal{D}_f$ (Yao et al., 2024a):

$$\mathcal{L} = \mathcal{L}_{\text{RLabel}}(\mathcal{D}_f). \tag{8}$$

Concretely, sample a random output y_{rdn} from a random-label distribution (for example, uniformly over a pre-constructed pool of irrelevant responses) and minimize the negative log-likelihood on these random targets:

$$\mathcal{L}_{\text{RLabel}}(\mathcal{D}_f) = \mathbb{E}_{(x,\cdot)\in\mathcal{D}_f}\,\mathbb{E}_{y_{\text{rdn}}}\big[-\log \pi_\theta(y_{\text{rdn}} \mid x)\big]. \tag{9}$$

These three families are widely used fine-tuning-based unlearning algorithms since they are classical, simple to implement, and easy to adapt to different forgetting objectives. Each family exhibits distinct strengths and limitations. GA and RLabel

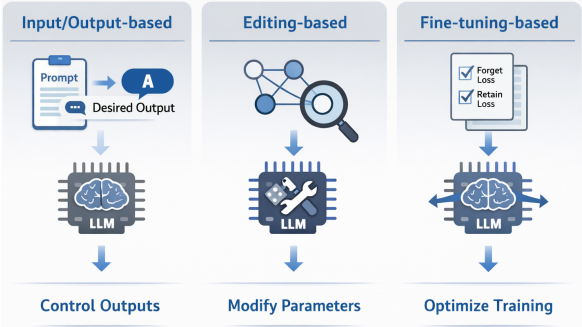

Figure 9: Three unlearning paradigms: input/output control without parameter changes, targeted parameter editing, and fine-tuning with forget–retain objectives.

can enforce forgetting effectively, but they may induce over-forgetting and degrade performance on the retain set (Yao et al., 2024a). NPO alleviates this issue by discouraging the model from assigning high likelihood to forget-set outputs (Zhang et al., 2024a), and retain-augmented variants such as NPO+GD and NPO+KL further help preserve utility. Notably, Tutek et al. (2025) uses unlearning as a *parametric intervention* to probe chain-of-thought faithfulness, and adopts NPO+KL as the underlying unlearning algorithm.

A.3 Evaluation

At the evaluation level, protocols have become increasingly comprehensive. Beyond average-case forgetting and utility, recent studies emphasize robustness under different attacks and practical constraints such as computation and runtime cost. We consolidate these references to facilitate verification and comparison, and to reduce the risk of overlooking contemporaneous work that may affect conclusions about coverage, methodological distinctions, or evaluation fairness. The taxonomy structure follows the main text, while this appendix prioritizes breadth and recency.

Some metrics naturally span multiple axes, and any discrete taxonomy may introduce mild ambiguity at the boundaries. For example, quantifying privacy leakage via membership inference, such as min-$k\%$-prob MIA AUC (Shi et al., 2024), can be viewed as task-level evaluation when treated as a target outcome reported alongside other end-task measures. It is also closely related to robustness, since it captures an adversary attempting to exploit training-set membership signals.

More broadly, the desiderata proposed in (Shi et al., 2025) include C1) no verbatim memorization, C2) no knowledge memorization, C3) no privacy leakage, C4) utility preservation, C5) scalability, and C6) sustainability, and these criteria cut across multiple components of our taxonomy. Concretely, ROUGE-style scores and other downstream accuracy measures are typically used for task-level utility, whereas MIA-style measurements reflect privacy or attack resilience and thus align with robustness-oriented evaluation.

Nevertheless, to keep the taxonomy navigable and reproducible, we assign each work to a primary category according to its dominant objective and experimental protocol. Some works naturally overlap across categories, but we expect such crossovers to be limited and not to affect the overall structure or conclusions. We hope this survey and taxonomy can help readers better understand the safety forgetting, quickly locate relevant benchmark, unlearning algorithm, and evaluation, and learn practical takeaways for future research and deployment.

B Details of Functional Forgetting

In this appendix, we provide more detailed information on the three functional forgetting directions: CoT faithfulness, learning with forgetting, and model regularizer.

B.1 CoT Faithfulness

Here, we first introduce chain-of-thought (CoT) and explain why it is central to behavior probing. CoT prompting, introduced by Wei et al. (2022), augments standard prompting with intermediate natural language reasoning steps, so that each exemplar takes the form of an ⟨input, chain of thought, output⟩ triple. In this setting, a CoT is a series of intermediate reasoning steps that lead to the final answer. Rather than directly producing only the final prediction, the model is encouraged to decompose a multi-step problem into smaller reasoning steps expressed in natural language.

This formulation is important for two reasons. First, CoT often improves performance on challenging reasoning tasks, especially in sufficiently LLMs (Wei et al., 2022). Second, it offers an interpretable view of model behavior by exposing a step-by-step rationale before the final answer. At the same time, CoT does not by itself establish that the model's internal computation faithfully follows the generated reasoning steps. In other words, a CoT may serve as a useful reasoning scaffold or externalized rationale, while its faithfulness to the model's actual decision process remains unclear.

Model	Method	Math (0-shot)	Math (4-shot)	GSM8K	MATH500	IFEval	TruthfulQA	MMLU	CSQA	All avg
Qwen2.5-1.5B	Original	7.10	31.38	66.34	1.40	32.13	46.65	59.64	74.61	39.91
	GA	8.48(+1.38)	35.58(+4.20)	66.94(+0.60)	5.60(+4.20)	37.05(+4.92)	47.19(+0.54)	59.68(+0.04)	74.61(+0.00)	**41.89**(+1.98)
	NPO	6.58(-0.52)	32.42(+1.04)	66.03(-0.31)	3.60(+2.20)	33.81(+1.68)	46.96(+0.31)	59.79(+0.15)	74.61(+0.00)	40.48(+0.57)
Qwen2.5-3B	Original	7.68	41.98	75.06	55.00	32.61	48.87	65.10	76.82	50.39
	GA	33.98(+26.30)	43.94(+1.96)	68.01(-7.05)	60.40(+5.40)	40.65(+8.04)	50.61(+1.74)	64.79(-0.31)	77.07(+0.25)	**54.93**(+4.54)
	NPO	11.66(+3.98)	45.00(+3.02)	72.48(-2.58)	55.60(+0.60)	36.21(+3.60)	49.99(+1.12)	65.09(-0.01)	76.82(+0.00)	51.61(+1.22)
Qwen2.5-7B	Original	9.00	51.42	80.10	57.80	43.17	56.31	71.93	85.42	56.89
	GA	50.44(+41.44)	67.22(+15.80)	79.61(-0.49)	62.40(+4.60)	51.56(+8.39)	58.33(+2.02)	71.57(-0.36)	84.85(-0.57)	**65.75**(+8.86)
	NPO	9.92(+0.92)	57.10(+5.68)	80.59(+0.49)	59.20(+1.40)	44.36(+1.19)	56.96(+0.65)	71.82(-0.11)	85.26(-0.16)	58.15(+1.26)
Qwen2.5-14B	Original	19.64	57.16	85.06	64.00	44.36	58.44	77.58	84.36	61.33
	GA	37.74(+18.10)	60.70(+3.54)	79.83(-5.23)	66.20(+2.20)	53.60(+9.24)	59.32(+0.88)	77.20(-0.38)	84.11(-0.25)	64.84(+3.51)
	NPO	47.56(+27.92)	62.16(+5.00)	85.75(+0.69)	67.60(+3.60)	49.04(+4.68)	59.48(+1.04)	77.48(-0.10)	84.36(+0.00)	**66.68**(+5.35)
Qwen2.5-7B-instruct	Original	48.74	67.74	84.15	72.80	73.00	64.70	71.75	82.72	70.70
	GA	55.02(+6.28)	70.26(+2.52)	86.88(+2.73)	74.21(+1.41)	75.06(+2.06)	61.00(-3.70)	71.44(-0.31)	81.49(-1.23)	71.92(+1.22)
	NPO	55.48(+6.74)	71.26(+3.52)	88.40(+4.25)	74.59(+1.79)	75.92(+2.92)	65.14(+0.44)	71.83(+0.08)	82.23(-0.49)	**73.11**(+2.41)
Qwen2.5-Math-7B	Original	13.74	53.24	86.66	62.10	29.74	48.25	57.93	62.24	51.74
	GA	15.20(+1.46)	56.96(+3.72)	87.49(+0.83)	63.27(+1.17)	32.25(+2.51)	48.53(+0.28)	57.83(-0.10)	62.24(+0.00)	**52.97**(+1.23)
	NPO	11.84(-1.90)	55.66(+2.42)	87.41(+0.75)	64.12(+2.02)	31.89(+2.15)	48.36(+0.11)	57.87(-0.06)	61.75(-0.49)	52.36(+0.62)
Llama-3.1-8B	Original	7.90	19.80	56.25	13.10	17.15	45.22	63.47	71.42	**36.79**
	GA	0.00(-7.90)	0.00(-19.80)	0.00(-56.25)	0.00(-13.10)	25.78(+8.63)	0.00(-45.22)	31.46(-32.01)	45.05(-26.37)	12.79(-24.00)
	NPO	7.66(-0.24)	18.20(-1.60)	0.45(-55.80)	0.89(-12.21)	10.79(-6.36)	43.99(-1.23)	63.45(-0.02)	71.09(-0.33)	27.07(-9.72)
Mistral-Small-24B-Base-2501	Original	6.24	41.70	81.73	71.20	23.62	52.89	76.91	74.37	**53.58**
	GA	0.00(-6.24)	0.00(-41.70)	0.00(-81.73)	0.00(-71.20)	24.58(+0.96)	0.00(-52.89)	26.89(-50.02)	19.25(-55.12)	8.84(-44.74)
	NPO	5.26(-0.98)	27.22(-14.48)	0.08(-81.65)	1.01(-70.19)	5.76(-17.86)	53.99(+1.10)	76.49(-0.42)	73.79(-0.58)	30.45(-23.13)

Table 6: Performance after unlearning NuminaMath-1.5 across different base models. GA and NPO denote Gradient Ascent and Negative Preference Optimization. For GA and NPO, each entry reports the score with the change in parentheses relative to the Original model within the same block. Positive and negative changes are shown in green and red, respectively. The best All avg score for each model is highlighted in bold.

This ambiguity motivates behavior probing via unlearning. If a CoT step is causally involved in producing the final answer, then suppressing that step should change the answer distribution. If the answer remains stable, the removed step is more likely to be post hoc rather than computationally necessary. From this perspective, unlearning can be viewed as a parametric intervention for testing whether a generated CoT reflects actual computation or merely surface-level rationalization.

B.2 Learning with Forgetting

As discussed in Section 4, the forget prompt set is first used to identify FC responses for removal, which are then collected into the unlearning set $\mathcal{D}_U$. The final training process follows a learning with forgetting scheme that alternates between learning and unlearning. Specifically, the model performs standard learning updates on $\mathcal{D}_L$, and after every N_u such updates, it executes one unlearning update on $\mathcal{D}_U$, for example via GA on the selected forget responses. In this way, the model learns from $\mathcal{D}_L$ while periodically suppressing responses in $\mathcal{D}_U$ that may hinder the learning objective. The resulting training dynamics can be written as the following periodic-batch objective.

$$\mathcal{L}_{pu}(x) = \sum_{x \in \{x_1^l, \ldots, x_{N_u}^l\}} \mathcal{L}(x) - \beta_{forget}\, \mathcal{L}(x^u),$$

where $\{x_1^l, \ldots, x_{N_u}^l\} \subset \mathcal{D}_L$, $x^u \in \mathcal{D}_U$, and β_{forget} controls the unlearning rate. This scheme suppresses obsolete information while reinforcing useful patterns, improving learning efficiency.

B.3 Model Regularizer

As a complement to the brief discussion in the main text, this appendix provides a fuller account of the motivation and case study details behind viewing unlearning as a potential regularization mechanism.

Recent progress in LLMs has led to steady gains on reasoning tasks such as mathematics (DeepSeek-AI et al., 2025; Chang et al., 2025; Yang et al., 2025; Dubey et al., 2024; Jaech et al., 2024). Post-training via SFT or RL often further improves performance, reinforcing the view that post-training plays a central role in shaping reasoning ability.

However, emerging evidence complicates this narrative. Several studies suggest that some reported gains may reflect pre-training contamination or memorization rather than genuine reasoning improvements. For example, applying RL to Qwen2.5-7B (Yang et al., 2024) with random or

even incorrect reward signals can still improve mathematical performance (Shao et al., 2025), casting doubt on reward optimization as the true causal driver. Similarly, as shown in Figure 6(c), Qwen2.5-7B can solve math problems from incomplete statements by implicitly reconstructing missing information, whereas other models cannot (Wu et al., 2025). These findings suggest that some post-training gains may be partly illusory, reflecting overfitting rather than true capability acquisition. From this perspective, machine unlearning may provide a useful tool for probing and potentially mitigating hidden memorization behaviors.

As a survey paper, we aim to highlight emerging trends rather than provide exhaustive experimentation. We therefore report representative studies and selected empirical evidence only to illustrate the patterns that motivate this perspective. Below, we describe the experimental setup and implementation details of the illustrative case study.

Experimental configuration. We evaluate whether unlearning can mitigate overfitting by selectively removing potentially overfitting-inducing data. Motivated by evidence that the Qwen2.5 series may be overfitted to mathematics-related tasks (Shao et al., 2025; Wu et al., 2025). We consider three model families, including the Qwen2.5 series (Yang et al., 2024), Llama-3.1-8B (Dubey et al., 2024), and Mistral-Small-24B-Base-2501[1]. For the forget set, we adopt NuminaMath-1.5, a recent dataset for mathematical reasoning (LI et al., 2024). We apply GA, NPO, and RLabel (Appendix A.2) to unlearn the NuminaMath-1.5 subset (LI et al., 2024), where we sample 10 subsets with 2,000 examples each, and then re-evaluate the resulting models under a fixed evaluation pipeline. In our experiments, RLabel only yields performance degradation, so we report results for GA and NPO in the main analysis [2].

All experiments use consistent settings and follow the optimizer configuration in (Touvron et al., 2023). We perform unlearning with AdamW (Loshchilov and Hutter, 2019), using a learning rate of 1.0×10^{-6}, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and $\epsilon = 10^{-8}$. We adopt a cosine learning rate schedule with a 10% warmup phase and decay to 10% of the peak rate. Weight decay is 0.1. All experiments are conducted on NVIDIA H100 GPUs.

Evaluation metrics. We evaluate both mathematical capability and general utility. Specifically, we report Math (0-shot and 4-shot) (Hendrycks et al., 2021b), and include GSM8K (Cobbe et al., 2021) and MATH500 (Hendrycks et al., 2021b) as standard math benchmarks. To assess broader instruction-following and generalization, we additionally report IFEval (Zhou et al., 2023), TruthfulQA (Lin et al., 2022), MMLU (Hendrycks et al., 2021a), and CommonsenseQA (CSQA) (Talmor et al., 2019), and summarize results with an overall average (All avg). We conduct all evaluations using LLM-Harness (Gao et al., 2024) and EvalChemy (Raoof et al., 2025).

Evaluation Result. The results are summarized in Table 6 and Figure 7. Overall, only the Qwen2.5 series shows consistent gains after unlearning NuminaMath-1.5, supporting the hypothesis that these models are more susceptible to math-related overfitting or memorization and thus more responsive to unlearning as a corrective update.

Within the Qwen2.5 family, Qwen2.5-7B exhibits the largest improvements. Both GA and NPO are beneficial, with GA often yielding larger gains, which is consistent with GA pushing parameters away from overspecialized regions and mitigating domain-specific overfitting. In contrast, Llama-3.1-8B and Mistral-Small-24B-Base-2501 do not benefit under the same setup, and unlearning instead leads to substantial degradation.

These mixed outcomes suggest that unlearning may function as a targeted regularizer when a model exhibits domain-specific overfitting, but it can be harmful when such a failure mode is absent. While our experiments are not intended to be exhaustive, they provide representative evidence that functional unlearning is a promising tool for probing and potentially reducing overfitting, and we hope this case study motivates more systematic investigations in future work.

C LLM Usage

We used ChatGPT as a writing assistant to improve language quality, refine phrasing, and enhance the overall readability of the manuscript.

[1] https://huggingface.co/mistralai/Mistral-Small-24B-Base-2501

[2] Our code is available at https://anonymous.4open.science/r/model_regularization-E88E.