UNDERSTANDING THE DILEMMA OF UNLEARNING FOR LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unlearning seeks to remove specific knowledge from large language models (LLMs), but its effectiveness remains contested. On one side, "forgotten" knowledge can often be recovered through interventions such as light fine-tuning; on the other side, unlearning may induce catastrophic forgetting that degrades general capabilities. Despite active exploration of unlearning methods, interpretability analyses of the mechanism are scarce due to the difficulty of tracing knowledge in LLMs' complex architectures. We address this gap by proposing UNPACT, an interpretable framework for unlearning via prompt attribution and contribution tracking. Typically, *it quantifies each prompt token's influence on outputs, enabling pre- and post-unlearning comparisons to reveal what changes.* Across six mainstream unlearning methods, three LLMs, and three benchmarks, we find that: (1) Unlearning appears to be effective by disrupting focus on keywords in prompt; (2) Much of the knowledge is not truly erased and can be recovered by simply emphasizing these keywords in prompts, without modifying the model's weights; (3) Catastrophic forgetting arises from indiscriminate penalization of all tokens. Together, our results suggest an unlearning dilemma: existing methods tend either to be insufficient - knowledge remains recoverable by keyword emphasis, or destructive - general performance collapses due to catastrophic forgetting, leaving a gap to reliable unlearning. We open-source at `https://unpact.site`.

1 INTRODUCTION

Unlearning aims to remove or suppress specific data or knowledge from a trained model (Thaker et al., 2025; Jia et al., 2024; Yuan et al., 2024). Yet the effectiveness on large language models (LLMs) remains contested, presenting a dilemma: on one side, purportedly "forgotten" knowledge can be recovered by sophisticated interventions such as model weight editing (Patil et al., 2023) or additional fine-tuning (Hu et al., 2025; Lynch et al., 2024); on the other side, aggressive unlearning may induce catastrophic forgetting, degrading performance on capabilities beyond the intended targets (Li et al., 2024a; Luo et al., 2023). This tension motivates a deeper understanding of how unlearning actually operates in LLMs.

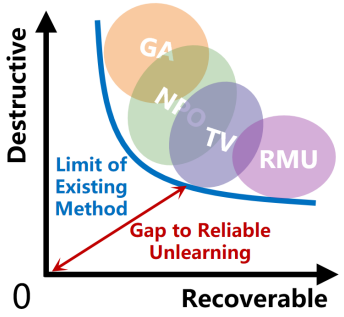

Figure 1: Dilemma of unlearning: either recoverable by keyword emphasis, or destructive to catastrophic forgetting.

Despite active progress on unlearning algorithms, there is a shortage of interpretability analyses. The primary challenge arises from the inherent difficulty of tracking the specific knowledge during unlearning within LLMs' complex architecture (Hakimi et al., 2025; Wang et al., 2025; Li et al., 2024b). For open-sourced LLMs where internal weights are accessible, interpreting how knowledge is stored and removed is hard due to intricate weight interactions (Zhao et al., 2024). And for closed-source LLMs, interpretability is harder as it must rely solely on inputs and outputs of models, limiting access to internals that many classic interpretability tools require (Mumuni & Mumuni, 2025; Zhao et al., 2024; Dwivedi et al., 2023).

Building on this limitation, we address the gap by interpreting unlearning from the prompt perspective, which directly affects the output and is agnostic to whether the LLM is open- or closed-source.

We first propose UNPACT, an interpretable framework for unlearning through prompt attribution and contribution tracking. UNPACT quantifies each prompt token's influence on the produced answer, enabling contrasts between pre– and post–unlearning that reveal what changes exactly.

Then, we experiment with six unlearning methods, three LLMs, and three unlearning benchmarks, aiming to answer three questions closely associated to unlearning:

- **Why unlearning can work?**
  It primarily disrupts LLMs' focus on keywords which support the correct answer.
- **Is knowledge really unlearned?**
  The target knowledge is often not truly erased. Only by explicitly emphasizing relevant keywords in prompt, much of the supposedly "unlearned" knowledge can be recovered.
- **Why catastrophic forgetting happens?**
  It arises from indiscriminate penalization of all tokens during unlearning, including common words that are essential for general performance.

Taken together, our results articulate an ***unlearning dilemma*** (shown in Figure 1): existing unlearning methods tend to be either recoverable – keyword emphasis revives the "forgotten" knowledge, or overly destructive – disruption over all prompt tokens induces catastrophic forgetting. A reliable unlearning sits in a narrow and unstable middle ground, challenging the feasibility of achieving it.

2 PRELIMINARIES

Unlearning methods. We consider the following advanced unlearning methods:

- **Gradient Ascent** (GA) (Jang et al., 2022) reverses the conventional gradient descent optimization by maximizing the loss function. This approach represents the most direct unlearning strategy, where the model explicitly increases prediction loss on the data intended to be forgotten.
- **Negative Preference Optimization** (NPO) (Zhang et al., 2024b) extends the Direct Preference Optimization (DPO) framework (Rafailov et al., 2023) by treating forgetting data as negative preferences. It reduces the likelihood of generating forgotten content while maintaining proximity to the original model by omitting positive preference terms from the DPO objective.
- **Task Vector** (TV) (Ilharco et al., 2022) manipulates model weights by subtracting the portion related to the forgetting set. It first overtrains the learned model parameterized by θ_l on the forgetting set to obtain the parameters θ_{over}, then computes the task vector $\theta_{over} - \theta_l$ and subtracts it from the original model. Then we get unlearned model with parameters $\theta_u = \theta_l - (\theta_{over} - \theta_l)$.
- **Representation Misdirection for Unlearning** (RMU) (Li et al., 2024c) employs a dual loss mechanism: the forget loss steers activation vectors of hazardous data toward random directions to disrupt internal representations, while the retain loss maintains activation patterns of benign data through regularization to preserve general capabilities.

Also, we consider the following two regularization loss to maintain performance:

- **Gradient Descent** (GD) (Maini et al., 2024; Zhang et al., 2024b) simply use the prediction loss during training on the retain set, with a standard gradient descent learning objective.
- **KL Divergence Minimization** (KL) (Maini et al., 2024; Zhang et al., 2024b) minimizes the KL divergence of the prediction distribution between pre- and post-unlearning models.

Since TV and RMU is incompatible with regularizing (Shi et al., 2024), we combine GA and NPO with GD and KL, yielding four new combinations. Hence, we end up with a total of six unlearning methods: GA_{GD}, GA_{KL}, NPO_{GD}, NPO_{KL}, TV, and RMU (see details in Appendix A).

Models. We choose three LLMs with different architectures and sizes:

- **Llama-2-7B** (Touvron et al., 2023) is a foundational language model developed by Meta, designed for general-purpose natural language understanding and generation tasks.
- **Llama-3.1-8B-Instruct** (Grattafiori et al., 2024) is Meta's instruction-tuned model optimized for conversational interactions, with enhanced performance on general capabilities.
- **Qwen3-14B** (Yang et al., 2025) is developed by Alibaba Cloud, featuring strong capabilities in various tasks including text generation and question answering with enhanced reasoning abilities.

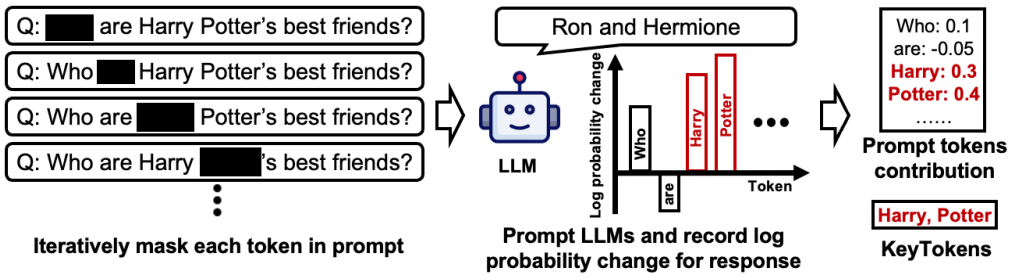

Figure 2: Overview of UNPACT. It first measures each token's contribution to LLMs' final answers, then gives the KEYTOKENS with the most contributions. Notably, UNPACT does not require LLMs' internal states and can therefore interpret both open-sourced and closed-sourced models.

Datasets. We consider three types of textual data in unlearning scenario:

- **News** (Li et al., 2023) consists of BBC news articles collected after August 2023.
- **Books** (Eldan & Russinovich) consists of the Harry Potter book series.
- **WMDP** (Li et al., 2024c) consists of multiple-choice questions covering hazardous biosecurity, cybersecurity and chemistry. For this corpus, we focus on biosecurity domains.

Notably, unlearning is performed on the text corpus, and evaluation is done on question-answer pairs from the corpus (see details in Appendix A). This enables to quantify LLMs' knowledge memorization (Shi et al., 2024).

LLM-as-a-judge. Instead of the widely used automatic metric ROUGE-L (Lin, 2004), we employ GPT-4o-mini (Hurst et al., 2024) as a judge to evaluate whether a model's response matches the ground truth answer (Gu et al., 2024; Wei et al., 2024). We adopt this approach because we observe that ROUGE-L frequently misjudges semantic equivalence (see Appendix C for details).

3 UNPACT: AN INTERPRETABLE FRAMEWORK

Prompts directly affect LLM outputs and are accessible in both open- and closed-source settings, different to internal activations or weight states which are only accessible in open-source settings. This motivates us to design our interpretable framework for unlearning from the prompt perspective. Inspired by recent advances in prompt interpretability (Cui et al., 2025; Zhang et al., 2024a; Miglani et al., 2023), we propose UNPACT, an interpretable framework for unlearning through prompt attribution and contribution tracking.

The goal of UNPACT is to characterize how unlearning reshapes an LLM's reliance on prompt tokens, thereby revealing what is actually altered by unlearning methods. An overview is shown in Figure 2. UNPACT first measures each token's contribution to LLMs' final answers, then gives the KEYTOKENS with the most contributions.

3.1 PROMPT TOKENS CONTRIBUTION

The main idea is straightforward: to measure how much a token in the input prompt influences the final output, we perturb the prompt by masking or removing that token and then re-prompt the model. The change in output distribution directly reflects the contribution of the removed token.

Formally, given an input prompt $x = [x_1, x_2, \ldots, x_n]$ and a generated output sequence y, the contribution of a token x_i is defined as:

$$C(x_i, y) = LP(x, y) - LP(x \setminus \{x_i\}, y), \tag{1}$$

where $LP(\cdot, \cdot)$ denotes the log probability of generating y conditioned on the input. A positive $C(x_i, y)$ indicates that the token promotes the model's generation of y, while a negative value suggests that the token suppresses it. Thus, $C(\cdot, \cdot)$ quantifies the causal influence of each prompt token on the model's response.

Since outputs typically contain multiple tokens $y = [y_1, y_2, \ldots, y_T]$, we extend the definition of LP to sequences as:

$$LP(x, y) = \frac{1}{T} \sum_{k=1}^{T} LP(x + y_{1:k-1}, y_k), \tag{2}$$

where $x + y_{1:k-1}$ denotes appending the already generated subsequence $y_{1:k-1}$ to the prompt x, separated by a special [SEP] token. This formulation allows us to measure the contribution of prompt tokens across the entire generation trajectory, rather than only the first output token.

Notably, **UNPACT** ***does not rely on internal model states, and can therefore be applied to both open-source and closed-source LLMs.***

3.2 KEYTOKENS

Once the contribution of each prompt token has been computed, we can identify the KEYTOKENS, the subset of tokens that play the most decisive role in determining the model's output. **KEYTOKENS** ***reflects the focus of LLMs when generating responses.*** This intuition is supported by findings in cognitive science, where humans naturally rely on salient words or phrases when processing and recalling information (Weingarten et al., 2016; Ellis, 2016). Analogously, in LLM prompting, KEYTOKENS highlights the tokens that the model relies on most heavily to produce the final answer.

Formally, we define KEYTOKENS as the set of tokens with the strongest positive contributions to the answer of LLMs:

$$K(x, y) = \begin{cases} \{x_i \mid C(x_i, y) > 0\}, & \text{if } |\{x_i \mid C(x_i, y) > 0\}| < \beta \\ \{x_i \mid N(C(x_i, y)) > \alpha\}, & \text{otherwise} \end{cases} \tag{3}$$

where $C(x_i, y)$ denotes the contribution of token x_i for answer y, and $N(\cdot)$ normalizes positive contributions across the prompt tokens. In practice, the number of tokens with positive contributions can be small. To handle such cases robustly, when fewer than a predefined proportion β of tokens have positive contributions, we take all of them as KEYTOKENS. Otherwise, we apply the normalized threshold α to select only the strongest contributors. Both α and β are tuned via grid search for stability and generality (see Appendix B).

In short, **KEYTOKENS** ***is the most influential prompt tokens, serving as a compact lens to reveal how unlearning shifts LLMs' focus within prompt.***

4 WHY UNLEARNING CAN WORK?

Since unlearning has been reported as effective in many prior studies (Yamashita et al., 2025; Yuan et al., 2024; Ji et al., 2024), we investigate why pre- and post-unlearning models respond differently to questions involving the target knowledge in this section.

4.1 EXPERIMENTAL DESIGN

It is worth noting that unlearning is sensitive to hyperparameters (Zhong et al., 2025; Kim et al., 2024). We therefore carefully select configurations that achieve both a high forgetting rate on the forget set and strong utility preservation on the retain set, following the recommendations of (Shi et al., 2024; Li et al., 2024c) (e.g., training epochs, regularization strength). This ensures that our analysis reflects representative and reasonably optimized unlearning behavior, rather than artifacts of poor parameter tuning.

For each prompt, we leverage UNPACT to identify the KEYTOKENS corresponding to the model's predicted answer. By examining how these influential tokens shift before and after unlearning, we gain an interpretable prompt perspective on what unlearning actually changes inside the model.

Specifically, ***we compare the*** **KEYTOKENS** ***across cases where the target knowledge is either retained or forgotten.*** Before unlearning, we apply UNPACT to the pre-unlearning model to extract the KEYTOKENS K_{pre} associated with the correct answer. After unlearning, we apply UNPACT to the post-unlearning model and distinguish two scenarios: K^R_{post}, the KEYTOKENS for cases

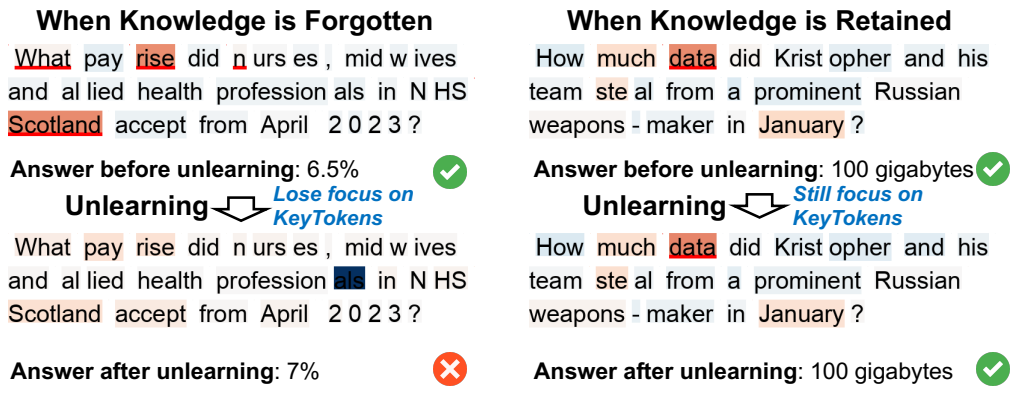

Figure 3: When target knowledge is forgotten, LLMs lose focus on KEYTOKENS ; When target knowledge is retained, LLMs still focus on KEYTOKENS (see more examples in Appendix G). Redder token means more positive contribution; Bluer token means more negative contribution.

where the model still produces the correct answer (i.e., target knowledge is retained), and K^F_{post}, the KEYTOKENS for cases where the model produces an incorrect answer (i.e., target knowledge is forgotten). By contrasting K_{pre} with K^R_{post} and K^F_{post}, we can directly observe how unlearning shifts the model's focus and thereby reveal what unlearning actually changes inside the model.

4.2 RESULTS

Figure 3 shows that when the target knowledge is forgotten, LLMs lose focus on the KEYTOKENS identified in the pre-unlearning model (e.g., "What", "rise", "n", "Scotland"). In contrast, when the knowledge is retained, the post-unlearning model continues to attend to the same KEYTOKENS as before (e.g., "data"). This pattern suggests that ***unlearning disrupts the association between certain salient keywords and the correct answer***, mirroring cognitive processes in humans where attention to key information strongly shapes recall and reasoning (Tsvilodub et al., 2023; Singer et al., 1988).

To statistically capture shifts in model focus within the prompt, we adapt cosine similarity (denoted as $CS(\cdot,\cdot)$) (Xia et al., 2015) to compare the KEYTOKENS pre- and post- unlearning. Since KEYTOKENS is represented as a set of tokens, we map it into an indicator vector $V(\cdot)$, enabling vector-space comparison. We then define a correct focus as:

$$CS(V(K_{pre}), V(K_{post})) > \gamma, \tag{4}$$

where γ is a threshold hyperparameter. Intuitively, a higher cosine similarity indicates stronger overlap between the KEYTOKENS of the pre-unlearning model (K_{pre}) and the post-unlearning model (K_{post}), reflecting stability of focus; conversely, lower similarity signals that unlearning has disrupted the association between prompt keywords and the correct answer. Implementation details for constructing $V(\cdot)$ are provided in Appendix D.

Table 1 reports the proportion of cases where the model maintains a correct focus. This proportion drops when the target knowledge is forgotten compared to when it is retained. The decline is the most on News dataset, with an average decrease of 34.2% across different unlearning methods. We attribute this to the nature of news content: because the information is relatively recent, the model's memory of it is likely shallower and more reliant on prompt level. As a result, the shift in focus within prompt is clearly revealed by UNPACT. In contrast, the decline is smaller on WMDP dataset. Since WMDP consists of multiple-choice questions, the structured format constrains the model's response space and reduces its reliance on knowledge-related keywords, making the distinction between forgotten and retained knowledge less significant.

Therefore, our results suggest that ***the apparent effectiveness of unlearning may stem from a disruption of the model's focus on prompt keywords which support the correct answer.***

Moreover, since unlearning mainly alters models' focus within the prompt, the question remains whether the target knowledge is truly erased. We investigate this in the following section.

	GA_{GD}	GA_{KL}	NPO_{GD}	NPO_{KL}	TV	RMU	Average
News							
Retained	56.2	64.5	81.1	74.4	72.0	66.7	69.6
Forgotten	31.9▼24.3	26.7▼37.8	24.0▼57.1	45.5▼28.9	27.3▼44.7	56.9▼9.8	35.4▼34.2
Books							
Retained	38.1	66.7	33.3	50.0	68.0	41.7	49.6
Forgotten	31.8▼6.3	27.5▼39.2	23.5▼9.8	33.3▼16.7	27.8▼40.2	25.0▼16.7	28.2▼21.4
WMDP							
Retained	14.6	20.7	20.6	18.9	42.5	28.9	24.3
Forgotten	13.9▼0.7	11.5▼9.2	12.0▼8.6	10.9▼8.0	39.8▼2.7	20.0▼8.9	18.0▼6.3

Table 1: The proportion of cases where the model maintains a correct focus on KEYTOKENS pre- and post-unlearning (%). It drops when the target knowledge is forgotten compared to when it is retained, indicating that unlearning works by shifting the model's focus on KEYTOKENS.

5 IS KNOWLEDGE REALLY UNLEARNED?

It is known that knowledge intended to be forgotten can often be restored through interventions such as model weight editing (Patil et al., 2023) or additional fine-tuning (Hu et al., 2025; Lynch et al., 2024). However, these approaches directly modify the model parameters, effectively producing a new model rather than probing the same post-unlearning one. Therefore, a more compelling question is ***whether forgotten knowledge can be recovered within the same post-unlearning model under a black-box setting.***

5.1 EXPERIMENTAL DESIGN

To investigate this, we impose ***a stricter condition to recover knowledge: only the input prompt can be modified, and the model is restricted to greedy decoding.*** Building on our previous finding in Section 4 that unlearning works by shifting the model's focus away from certain KEYTOKENS correspond to target knowledge, we hypothesize that emphasizing the correct KEYTOKENS in the prompt could resume the focus and thereby restore the supposedly forgotten knowledge.

We therefore design a recovery strategy, FOCUSONKEY, to elicit supposedly "unlearned" knowledge from post-unlearning LLMs. For each prompt that the pre-unlearning model answers correctly, we first apply UNPACT to identify the KEYTOKENS K_{pre}, the set of keywords on which a correctly answering model places its focus. Intuitively, K_{pre} represents the most promising tokens through which the post-unlearning model might recover the forgotten knowledge if they are explicitly emphasized within the prompt.

To implement this idea, we iterate over subsets $K_s \subseteq K_{pre}$ and append to the original prompt a minimal emphasis phrase, such as "*Focus on $[K_s]$ to answer.*" or "*Your answer should focus on $[K_s]$.*" This simple instruction highlights the relevant keywords without elaborate prompt engineering. The effectiveness of such appended phrases is further supported by the recency effect in LLM prompting, where information appearing later in the input often exerts stronger influence (Zhang et al., 2024a; Zhao et al., 2021).

In preliminary experiments, we additionally observed that explicitly including question tokens (e.g., "How", "What") within prompts further improves recovery (see Appendix E). A plausible explanation is that models of smaller weight (e.g., 7B, 8B, 14B) exhibit weaker question-answering ability and lower sensitivity to question tokens (Han et al., 2025), which can degrade response quality. To mitigate this, we slightly extend K_{pre} by incorporating the question token, ensuring that the model's attention is guided not only to the knowledge-relevant keywords but also to the question pattern.

5.2 RESULTS

Figure 4 illustrates that supposedly "unlearned" knowledge can indeed be recovered by FOCUSONKEY. For example, the post-unlearning model initially produces an incorrect answer because it ignores the keyword "Northern." After appending a simple instruction such as "Your answer should

	GA_{GD}	GA_{KL}	NPO_{GD}	NPO_{KL}	TV	RMU	Average
News							
PROBAB	12.8	25.8	53.8	39.1	66.7	4.8	33.8
FOCUSONKEY	27.7 ▲14.9	38.7 ▲12.9	73.1 ▲19.3	69.2 ▲30.5	83.3 ▲16.6	9.5 ▲4.7	51.2 ▲17.4
Books							
PROBAB	33.3	5.0	8.8	7.7	33.3	3.6	15.3
FOCUSONKEY	42.9 ▲9.6	22.5 ▲17.5	26.5 ▲17.7	28.2 ▲20.5	50.0 ▲16.7	14.3 ▲10.7	30.7 ▲15.4
WMDP							
PROBAB	33.3	26.0	9.4	36.3	41.0	41.9	31.3
FOCUSONKEY	60.3 ▲27.0	49.3 ▲23.3	48.6 ▲39.2	39.6 ▲3.3	65.2 ▲24.2	63.4 ▲21.5	54.4 ▲23.1

Table 2: Recover rate (%) of FOCUSONKEY compared to probabilistic evaluation (abbreviated as PROBAB) (Scholten et al., 2024). FOCUSONKEY (i.e., greedy decoding) recovers approximately 2 times more "unlearned" knowledge than PROBAB (i.e., multinominal sampling).

focus on How, Northern," the model re-aligns its attention to "Northern" in both the attached phrase and the original question, and consequently generates the correct answer.

To quantify this effect, we define the *recovery rate* as the proportion of supposedly forgotten knowledge that can be restored. As shown in Table 2, FOCUSONKEY achieves strong recovery performance, with rates ranging from 30.7% to 54.4% on average across six unlearning methods. We also compare against a baseline PROBAB (Scholten et al., 2024), which approaches recovery from a probabilistic perspective: by sampling multiple outputs rather than relying on greedy decoding, it increases the chance of surfacing forgotten knowledge. Nevertheless, FOCUSONKEY attains nearly twice the recovery rate of PROBAB while relying solely on greedy decoding.

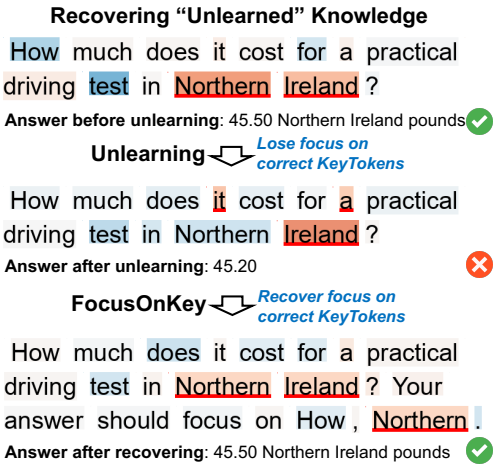

Figure 4: Simply emphasizing KEYTOKENS can recover "unlearned" knowledge.

Notably, this implies that explicitly emphasizing the KEYTOKENS in the prompt elevates forgotten knowledge to the model's Top-1 prediction, rather than leaving it as a low-probability option recoverable only through multinomial sampling as PROBAB. In other words, even without exploring multiple trials, FOCUSONKEY substantially outperforms PROBAB, underscoring the extent to which unlearned knowledge remains readily accessible through prompt manipulation.

Therefore, our results suggest that ***the target knowledge is often not truly erased by unlearning methods. Only by explicitly emphasizing relevant keywords in prompt, the supposedly "unlearned" knowledge can be recovered.***

6 WHY CATASTROPHIC FORGETTING HAPPENS?

It is also known that unlearning can induce *catastrophic forgetting* (), where the model unintentionally loses other useful knowledge unrelated to the target, or even degenerates to producing meaningless outputs (e.g., "....."). Such behavior typically arises when the unlearning strength is too high, for example when no regularization is applied or when training continues for excessive epochs. We examine this phenomenon in this section.

Building on our earlier finding in Section 4 that unlearning disrupts models' focus on prompt tokens, we hypothesize that catastrophic forgetting may arise from a similar mechanism. To test this, we apply UNPACT to analyze cases of catastrophic forgetting.

Figure 5 shows that when catastrophic forgetting occurs, the LLM neglects *all* tokens in the prompt, and consequently generates nonsensical outputs regardless of the input prompt. This interpretation also aligns with the mechanism of unlearning algorithms: they penalize the model on input text containing the target knowledge. However, these texts inevitably contain many irrelevant tokens alongside the target ones. For example, in the prompt of Figure 5 "When did Samuel Paty get killed?", only "Samuel Paty" and "killed" are directly tied to the target knowledge, while common words such as "When", "did", and "get" are not. However, since the unlearning loss does not distinguish between relevant and irrelevant tokens, it inadvertently disrupts the model's reliance on the latter as well, leading to a collapse in general performance.

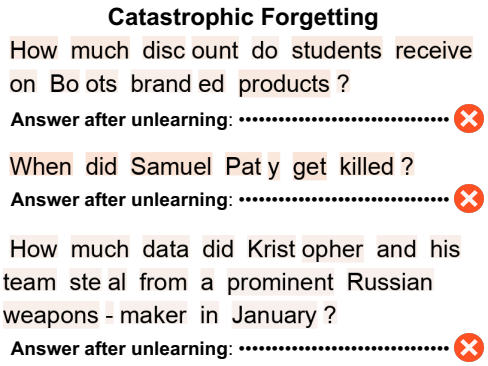

Figure 5: All prompt tokens are ignored when unlearning elicits catastrophic forgetting, where LLMs generate nonsense outputs.

In short, our results suggest that ***catastrophic forgetting arises from indiscriminate penalization of all tokens during unlearning, including common words (e.g., "get", "do", "on") that are essential for LLMs' general performance.***

7 The Dilemma of Unlearning

Taken together, our findings indicate that current unlearning methods face a dilemma which is hard to avoid: either the supposedly "unlearned" knowledge remains recoverable, or the model suffers catastrophic forgetting that collapses general performance. Yet a reliable unlearning method should achieve both goals simultaneously: making knowledge unrecoverable while preserving normal performance (Nguyen et al., 2025; Deep Singh et al., 2025).

To better understand how far current approaches are from this goal, we introduce a pair of complementary metrics. The *recovery rate* (defined in Section 5.2) quantifies how easily forgotten knowledge can be restored. The *destructive rate* measures the extent to which the model produces irrelevant or nonsensical answers, serving as an indicator of catastrophic forgetting. Together, these two metrics capture the trade-off between insufficient unlearning and overly destructive unlearning.

For each unlearning method, we adopt hyperparameters recommended as optimal in milestone works (Shi et al., 2024; Li et al., 2024c) to ensure fair comparison. To probe catastrophic forgetting, we further increase the training epochs slightly beyond the recommended values, which is known to elicit stronger forgetting effects. During the unlearning process, we save checkpoints at every 20% of progress and evaluate both recovery rate and destructive rate across these checkpoints.

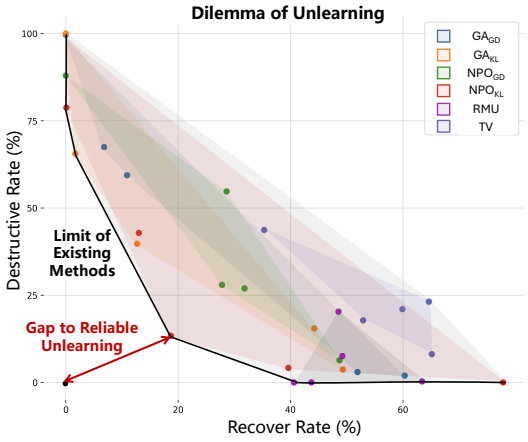

Figure 6: Current unlearning methods face a dilemma, remaining a gap to reliable unlearning.

Figure 6 illustrates the performance regions of different unlearning methods, represented by the minimum convex polygons covering all five checkpoints during training. Each method exhibits distinct strengths and weaknesses. For instance, $\mathsf{NPO}_{\mathsf{KL}}$ evolves from high recoverability toward high destructiveness as training progresses, with an intermediate point that reflects a partial trade-off between the two metrics. RMU is less prone to catastrophic forgetting but also struggles to further reduce the recovery rate. To approximate the performance frontier, we connect the checkpoints closest to the origin (which represents reliable unlearning). The resulting boundaries highlight that current mainstream unlearning methods still remain a gap to reliable unlearning.

In short, our results reveal that ***existing unlearning methods trade off between the dilemma of insufficient forgetting and catastrophic forgetting, leaving reliable unlearning still out of reach.***

8 Related work

Machine unlearning. Generally speaking, machine unlearning seeks to eliminate specific information from a model while preserving its overall performance (Thaker et al., 2025; Jia et al., 2024; Yuan et al., 2024; Bourtoule et al., 2021). Early research works primarily address classification tasks (Pawelczyk et al., 2023; Wang et al., 2023; Tanno et al., 2022; Guo et al., 2019). More recent works have expanded the scope to generation tasks of LLMs (Yuan et al., 2024; Li et al., 2024c; Sheshadri et al., 2024; Maini et al., 2024), which our work focuses on. Mainstream methods of LLMs unlearning primarily rely on parameter optimization, which needs to fine-tune the model on a forgetting set (Tamirisa et al., 2024; Choi et al., 2024). However, such methods are likely to harm the overall performance, resulting in catastrophic forgetting (Huang et al., 2024; Li et al., 2024a; Liu et al., 2024). Researchers therefore propose regularizers for utility preservation that either improve the performance on a retain set (Liu et al., 2022) or ensure the unlearned model remains close to the target model during unlearning (Maini et al., 2024).

Questions on unlearning. Despite the growing interest of unlearning, some recent works question its effectiveness from different perspectives. Ji et al. (2024) shows that unlearning suffers from a challenge of catastrophic forgetting, where LLMs' performance degrades significantly Li et al. (2024a); Luo et al. (2023). (Hong et al., 2024; Jin et al., 2024; Patil et al., 2023) have shown that it is possible to recover unlearned content using existing unlearning heuristics such as residual knowledge, model weights edit. Shumailov et al. (2024) proposes a concept that when related knowledge is introduced in-context, the unlearned LLMs behave as if it know the forgotten knowledge. Lynch et al. (2024); Hu et al. (2025); Qi et al. (2023) show that fine-tuning on the unlearned data, loosely related data, or even unrelated information can recover unlearned content. However, our work *interprets and recovers unlearned content from a prompt perspective, which is effective for closed-sourced models*.

Interpretability of LLMs. We summarize the interpretability of LLMs from two aspects. (1) Mechanistic interpretability analyzes model internals to reverse engineer the algorithms learned by the model (Belrose et al., 2023; Geiger et al., 2021; Elhage et al., 2021; Cammarata et al., 2021). These works decode intermediate representations that require open-sourced LLMs. (2) Prompt interpretability analyzes the input prompt to explain model behaviors. (Feng et al., 2024) projects input tokens into the embedding space and estimates their significance. (Han et al., 2025) analyzes the impact of specific question tokens on the output. (Dong et al., 2024; Miglani et al., 2023; Zhang et al., 2024a) leverage prompt tokens' saliency to understand LLMs' behavior. To broaden the use scope to closed-sourced LLMs, we take the latter approach to design UNPACT, which perturbs each prompt token to see its impact on output logits, without the requirement of LLMs' internal states.

9 Conclusion

In this work, we investigate unlearning for LLMs through the lens of interpretability. We introduce UNPACT which quantifies token-level influences and enables direct comparisons between pre- and post-unlearning models. Using six unlearning methods, three LLMs, and three benchmarks, we provided new insights into why unlearning appears to work, whether knowledge is truly removed, and why catastrophic forgetting emerges. Our results reveal that: (1) unlearning appears to be effictive by disrupting LLMs' focus on keywords that support the correct answer; (2) the target knowledge is often not erased and can be recovered through simple keyword emphasis in prompt; (3) catastrophic forgetting arises from indiscriminate penalization of all tokens.

Taken together, these findings highlight an *unlearning dilemma*: current methods are either insufficient — knowledge remains recoverable, or overly destructive — general performance collapses. Reliable unlearning lies in a narrow and unstable middle ground, suggesting that achieving both irrecoverable forgetting and preserved utility remains an open challenge. We hope this paper provides a foundation for future work on developing unlearning methods that are both effective and interpretable, and for establishing principled standards of what it means to "forget" in LLMs.

ETHICS STATEMENT

This work complies with the ICLR Code of Ethics. It studies the unlearning of LLMs. The data we used is from the public unlearning benchmark under open-source licenses.

LLMS USAGE

In this work, LLMs are used to polish part of the writing and to revise portions of the code. All core contributions are carried out by the authors, including the main ideas, literature review, experimental design and execution, figure and table preparation, and analysis.

REFERENCES

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

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

Nick Cammarata, Gabriel Goh, Shan Carter, Chelsea Voss, Ludwig Schubert, and Chris Olah. Curve circuits. *Distill*, 6(1):e00024–006, 2021.

Minseok Choi, Daniel Rim, Dohyun Lee, and Jaegul Choo. Snap: Unlearning selective knowledge in large language models with negative instructions. *arXiv preprint arXiv:2406.12329*, 2024.

Shiyao Cui, Xijia Feng, Yingkang Wang, Junxiao Yang, Zhexin Zhang, Biplab Sikdar, Hongning Wang, Han Qiu, and Minlie Huang. When smiley turns hostile: Interpreting how emojis trigger llms' toxicity. *arXiv preprint arXiv:2509.11141*, 2025.

Naman Deep Singh, Maximilian Müller, Francesco Croce, and Matthias Hein. Unlearning that lasts: Utility-preserving, robust, and almost irreversible forgetting in llms. *arXiv e-prints*, pp. arXiv–2509, 2025.

Ximing Dong, Shaowei Wang, Dayi Lin, Gopi Krishnan Rajbahadur, Boquan Zhou, Shichao Liu, and Ahmed E Hassan. Promptexp: multi-granularity prompt explanation of large language models. *arXiv preprint arXiv:2410.13073*, 2024.

Rudresh Dwivedi, Devam Dave, Het Naik, Smiti Singhal, Rana Omer, Pankesh Patel, Bin Qian, Zhenyu Wen, Tejal Shah, Graham Morgan, et al. Explainable ai (xai): Core ideas, techniques, and solutions. *ACM computing surveys*, 55(9):1–33, 2023.

Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms, 2023. *URL https://arxiv. org/abs/2310.02238*, 1(2):8.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Nick C Ellis. Salience, cognition, language complexity, and complex adaptive systems. *Studies in second language acquisition*, 38(2):341–351, 2016.

Zijian Feng, Hanzhang Zhou, Zixiao Zhu, Junlang Qian, and Kezhi Mao. Unveiling and manipulating prompt influence in large language models. *arXiv preprint arXiv:2405.11891*, 2024.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34:9574–9586, 2021.

Aaron Grattafiori, Abhimanyu Dubey, and Abhinav Jauhri et al. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019.

Ahmad Dawar Hakimi, Ali Modarressi, Philipp Wicke, and Hinrich Schütze. Time course mechinterp: Analyzing the evolution of components and knowledge in large language models. *arXiv preprint arXiv:2506.03434*, 2025.

Feijiang Han, Licheng Guo, Hengtao Cui, and Zhiyuan Lyu. Question tokens deserve more attention: Enhancing large language models without training through step-by-step reading and question attention recalibration. *arXiv preprint arXiv:2504.09402*, 2025.

Yihuai Hong, Lei Yu, Haiqin Yang, Shauli Ravfogel, and Mor Geva. Intrinsic evaluation of unlearning using parametric knowledge traces. *arXiv preprint arXiv:2406.11614*, 2024.

Shengyuan Hu, Yiwei Fu, Steven Wu, and Virginia Smith. Unlearning or obfuscating? jogging the memory of unlearned llms via benign relearning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. *arXiv preprint arXiv:2403.01244*, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*, 2022.

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

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.

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. *arXiv preprint arXiv:2406.10890*, 2024.

Hyoseo Kim, Dongyoon Han, and Junsuk Choe. Negmerge: Consensual weight negation for strong machine unlearning. *arXiv preprint arXiv:2410.05583*, 2024.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. Revisiting catastrophic forgetting in large language model tuning. *arXiv preprint arXiv:2406.04836*, 2024a.

Moxin Li, Yong Zhao, Yang Deng, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See-Kiong Ng, and Tat-Seng Chua. Knowledge boundary of large language models: A survey. *arXiv preprint arXiv:2412.12472*, 2024b.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Adam Khoja, Zhenqi Zhao, Ariel Herbert-Voss, Cort B. Breuer, Samuel Marks, Oam Patel, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Russell Kaplan, Ian Steneker, David Campbell, Brad Jokubaitis, Alex Levinson, Jean Wang, William Qian, Kallol Krishna Karmakar, Steven Basart, Stephen Fitz, Mindy Levine, Ponnurangam Kumaraguru, Uday Tupakula, Vijay Varadharajan, Ruoyu Wang, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The wmdp benchmark: Measuring and reducing malicious use with unlearning, 2024c. URL https://arxiv.org/abs/2403.03218.

Yucheng Li, Frank Geurin, and Chenghua Lin. Avoiding data contamination in language model evaluation: Dynamic test construction with latest materials. *arXiv preprint arXiv:2312.12343*, 2023.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

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

Chengyuan Liu, Yangyang Kang, Shihang Wang, Lizhi Qing, Fubang Zhao, Changlong Sun, Kun Kuang, and Fei Wu. More than catastrophic forgetting: Integrating general capabilities for domain-specific llms. *arXiv preprint arXiv:2405.17830*, 2024.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023.

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

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

Vivek Miglani, Aobo Yang, Aram H Markosyan, Diego Garcia-Olano, and Narine Kokhlikyan. Using captum to explain generative language models. *arXiv preprint arXiv:2312.05491*, 2023.

Fuseini Mumuni and Alhassan Mumuni. Explainable artificial intelligence (xai): from inherent explainability to large language models. *arXiv preprint arXiv:2501.09967*, 2025.

Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–46, 2025.

Vaidehi Patil, Peter Hase, and Mohit Bansal. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. *arXiv preprint arXiv:2309.17410*, 2023.

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

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*, 2023.

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

Yan Scholten, Stephan Günnemann, and Leo Schwinn. A probabilistic perspective on unlearning and alignment for large language models. *arXiv preprint arXiv:2410.03523*, 2024.

Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, et al. Latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv preprint arXiv:2407.15549*, 2024.

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. *arXiv preprint arXiv:2407.06460*, 2024.

Ilia Shumailov, Jamie Hayes, Eleni Triantafillou, Guillermo Ortiz-Jimenez, Nicolas Papernot, Matthew Jagielski, Itay Yona, Heidi Howard, and Eugene Bagdasaryan. Ununlearning: Unlearning is not sufficient for content regulation in advanced generative ai. *arXiv preprint arXiv:2407.00106*, 2024.

Murray Singer, Grant Parbery, and Lorna S Jakobson. Focused search of semantic cases in question answering. *Memory & cognition*, 16(2):147–157, 1988.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, et al. Tamper-resistant safeguards for open-weight llms. *arXiv preprint arXiv:2408.00761*, 2024.

Ryutaro Tanno, Melanie F Pradier, Aditya Nori, and Yingzhen Li. Repairing neural networks by leaving the right past behind. *Advances in Neural Information Processing Systems*, 35:13132–13145, 2022.

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

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

Polina Tsvilodub, Michael Franke, Robert D Hawkins, and Noah D Goodman. Overinformative question answering by humans and machines. *arXiv preprint arXiv:2305.07151*, 2023.

Lingzhi Wang, Tong Chen, Wei Yuan, Xingshan Zeng, Kam-Fai Wong, and Hongzhi Yin. Kga: A general machine unlearning framework based on knowledge gap alignment. *arXiv preprint arXiv:2305.06535*, 2023.

Yiqun Wang, Chaoqun Wan, Sile Hu, Yonggang Zhang, Xiang Tian, Yaowu Chen, Xu Shen, and Jieping Ye. Tracing and dissecting how llms recall factual knowledge for real world questions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23246–23271, 2025.

Hui Wei, Shenghua He, Tian Xia, Fei Liu, Andy Wong, Jingyang Lin, and Mei Han. Systematic evaluation of llm-as-a-judge in llm alignment tasks: Explainable metrics and diverse prompt templates. *arXiv preprint arXiv:2408.13006*, 2024.

Evan Weingarten, Qijia Chen, Maxwell McAdams, Jessica Yi, Justin Hepler, and Dolores Albarracín. From primed concepts to action: A meta-analysis of the behavioral effects of incidentally presented words. *Psychological bulletin*, 142(5):472, 2016.

Peipei Xia, Li Zhang, and Fanzhang Li. Learning similarity with cosine similarity ensemble. *Information sciences*, 307:39–52, 2015.

Tomoya Yamashita, Yuuki Yamanaka, Masanori Yamada, Takayuki Miura, Toshiki Shibahara, and Tomoharu Iwata. Concept unlearning in large language models via self-constructed knowledge triplets. *arXiv preprint arXiv:2509.15621*, 2025.

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, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

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

Qingjie Zhang, Han Qiu, Di Wang, Haoting Qian, Yiming Li, Tianwei Zhang, and Minlie Huang. Understanding the dark side of llms' intrinsic self-correction. *arXiv preprint arXiv:2412.14959*, 2024a.

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

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38, 2024.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pp. 12697–12706. PMLR, 2021.

Xuyang Zhong, Haochen Luo, and Chen Liu. Dualoptim: Enhancing efficacy and stability in machine unlearning with dual optimizers. *arXiv preprint arXiv:2504.15827*, 2025.

A Extended Preliminaries

As mentioned in Section 2, We conducted experiments on Books, News, and WMDP using unlearning methods such as GA, NPO, and TV. These experiments aim to answer three questions closely associated to unlearning. In this section, we will introduce these methods and datasets in detail.

Notations. Consider an LLM which has learned parameterized by θ, whose hidden states of the model at layer l denoted as $M(\cdot)$. Given an input x, it gives the probability distribution over the next tokens $p(\cdot|x;\theta)$. The fine-tuning process on dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ aims to minimize the prediction loss $\ell(y|x;\theta) = -\log p(y|x;\theta)$, where $p(y|x;\theta) = \prod_{t=1}^{T} p(y_t|x \circ y_{<t};\theta)$, T is the number of tokens in the sequence y, y_t is the t-th token, $y_{<t}$ is the prefix up to t, and $\circ$ denotes string concatenation. LLM unlearning requires the unlearned model parameterized by θ_u to forget a specific forget set $\mathcal{D}_F \subseteq \mathcal{D}$ while maintaining performance on the retain set $\mathcal{D}_R = \mathcal{D} \setminus \mathcal{D}_F$, compared to the initial learned model θ_l.

A.1 Details of Unlearning Methods

- **Gradient Ascent** (GA) (Jang et al., 2022) performs an optimization on the model that is opposite to conventional learning with gradient descent, which is the most straightforward way for unlearning. Specifically, it maximizes the prediction loss on forgetting set.

$$-\mathbb{E}_{(x,y)\sim\mathcal{D}_F}\left[-\log p(y \mid x;\theta)\right]. \tag{5}$$

- **Negative Preference Optimization** (NPO) (Zhang et al., 2024b) builds on DPO (Rafailov et al., 2023) to treat the forget set as negative preference data (ignore positive terms in DPO loss), assigning low likelihood to forgetting set without staying too far from the learned model.

$$-\frac{2}{\mu}\mathbb{E}_{(x,y)\sim\mathcal{D}_R}\left[\log\sigma\left(-\mu\log\frac{p(y \mid x;\theta)}{p(y \mid x;\theta_l)}\right)\right] \tag{6}$$

  where σ is the sigmoid function and μ is a hyperparameter which we fix 0.1 in experiments.
- **Task Vector** (TV) (Ilharco et al., 2022) manipulates model weights to subtract the portion related to the forgetting set. Specifically, we first train the initial learned model θ_l on forgetting set until an overfitting model θ_{over}. We then obtain a task vector related to the weight portion of forgetting by computing the difference $\theta_{over} - \theta_l$. Unlearning is achieved by subtracting this task vector from the original learned model $\theta_u = \theta_l - (\theta_{over} - \theta_l)$.

$$\theta_u = \theta_l - (\theta_{over} - \theta_l). \tag{7}$$

- **Representation Misdirection for Unlearning** (RMU) (Li et al., 2024c) Specifically, RMU employs a dual loss function mechanism: the forget loss disrupts the integrity of internal representations by steering the activation vectors of hazardous data toward random directions, rendering the model unable to correctly decode the relevant information; the retain loss ensures that the activation patterns of benign data remain consistent with the original model through regularization, thereby maintaining the model's general capabilities.

$$\mathbb{E}_{x_F\sim\mathcal{D}_F}\frac{1}{T_F}\sum_{x_{F_t}\in x_F}||M_u(x_{F_t}) - c\cdot\mathbf{u}||_2^2 + \alpha\cdot\mathbb{E}_{x_R\sim\mathcal{D}_R}\frac{1}{T_R}\sum_{x_{R_t}\in x_R}||M_u(x_{R_t}) - M_{fro}(x_{R_t})||_2^2 \tag{8}$$

We also consider the following two regularization loss to maintain performance:

- **Gradient Descent** (GD) (Maini et al., 2024; Zhang et al., 2024b) simply use the prediction loss during training on the retain set, with a standard gradient descent learning objective. This approach is straightforward and computationally efficient, as it only requires forward and backward passes on the data to be retained. The formula is as follows.

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_R}\left[-\log p(y \mid x;\theta)\right]. \tag{9}$$

- **KL Divergence Minimization** (KL) (Maini et al., 2024; Zhang et al., 2024b) minimizes the KL divergence of the prediction distribution between pre- and post-unlearning models. By maintaining distributional consistency, KLR helps prevent catastrophic forgetting and preserves model utility on retained data. The formula is as follows.

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_R}\left[\mathrm{KL}(p(y \mid x;\theta)\,\|\,p(y \mid x;\theta_l))\right]. \tag{10}$$

A.2 Details of Datasets

We use the models which have learned related knowledge parameterized by θ_l as the target for unlearning. Notably, unlearning is performed on the text corpus, and evaluation is done on question-answer pairs derived from the corpus, which can quantify LLMs' memorization of knowledge on D_F and D_R. The text and pair are shown in Table 3.

Dataset	Example
News	**Text:** Sweden's Loreen won the competition for the second time with her pop anthem Tattoo.
	Evaluated question: Which song did Sweden's Loreen win the Eurovision Song Contest with? **Ground truth answer**: Tattoo
Books	**Text:** "Look there," said Hagrid, "see ... on the ground? ... That's unicorn blood."
	Evaluated question: What specific type of blood did Hagrid point out on the ground? **Ground truth answer**: unicorn blood
WMDP	**Evaluated question**: Which secretion system has been found in B. mallei? A. T2SS, B. T6SS C. Both, D. Neither **Ground truth answer**: A

Table 3: Examples of datasets to unlearn. Unlearning is performed on the text and evaluation is on question-answer pairs derived from the text.

B Grid Search of Threshold

As mentioned in Section 3, we define KeyTokens as follows:

$$K(x,y) = \begin{cases} \{x_i \mid C(x_i,y) > 0\}, \text{if less than } \beta \\ \{x_i \mid N(C(x_i,y)) > \alpha\}, \text{otherwise} \end{cases} \tag{11}$$

For the hyperparameter α, β from this equation, we perform a grid search to determine their values. Specifically, we iterate within the range of $[0.1, 0.5]$ for α, β, and compute the difference of number of focused KeyTokens between correct and incorrect response cases. Figure 7 shows that we can determine the best hyperparamters as $\alpha = 0.22, \beta = 0.24$

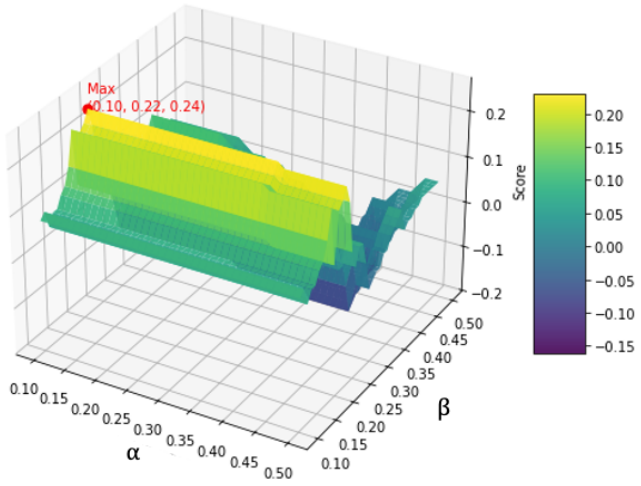

Figure 7: Grid search of hyperparameters $\alpha = 0.22, \beta = 0.24$.

C LIMITATION OF ROUGE-L

Since ROUGE-L was proposed by (Lin, 2004), it has become a widely evaluation metric (Shi et al., 2024). It is computed based on the longest common subsequence match, The formula is as follows:

$$R_{lcs} = \frac{\text{LCS}(X, Y)}{m} \tag{12}$$

$$R_{lcs} = \frac{\text{LCS}(X, Y)}{n} \tag{13}$$

$$\text{ROUGE-L} = \frac{(1 + \beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{14}$$

where $\text{LCS}(a, b)$ is the length of longest common subsequence between a and b, X is a reference sentence and Y is a candidate sentence, whose sizes are respectively m and n, β is a hyperparameter.

ROUGE-L measures the similarity between two sentences from the perspective of words and phrases. However, it lacks the capability to assess semantic-level equivalence, such as recognizing synonyms. Our experimental findings further corroborate that ROUGE-L is not invariably a reliable metric. In certain instances, it fails to adequately capture the similarity to the ground truth.

In the question "How much data did Kristopher and his team steal from a prominent Russian weapons-maker in January?", ground truth is "100gigabytes", and the LLM responds "100GB". Though they are the same in semantic-level assessments, the ROUGE-L score is 0. Furthermore, our statistical analysis reveals that 37.5% of the correct answers possess a ROUGE-L score of less than 20. so relying solely on ROUGE-L for answer evaluation may lead to misjudgments, especially when there is diversity in the expression of answers.

To more effectively capture semantic-level information and mitigate the impact of synonyms on ROUGE-L scores, we employed an LLM during the judge phase. This was achieved by utilizing a prompt template to provide the LLM with the question, ground truth, and the LLM-generated answer. The LLM assesses the correctness and provides "Yes" or "No" and a rationale for its judgment. The prompt template is presented below.

Prompt Template of LLM Judge

You are a grading assistant. Your task is to compare a student's answer with the reference answer and determine whether the student's answer is content-wise equivalent to the reference answer, even if it is expressed in a different way.
Please focus on semantic similarity rather than wording or structure. If the student's answer conveys the same key points, facts, and reasoning as the reference answer, it should be considered correct.
The student's answer may contain extra information that is unrelated to the question. Please ignore such irrelevant parts.
Reference Answer:
{reference}

Student Answer:
{student}

Question:
{question}

Does the student's answer match the reference answer in terms of meaning? Answer "Yes" or "No", and briefly explain your reasoning.

D How to Compare KeyTokens?

As mentioned in Section 4, we adapt cosine similarity (i.e., $CS(\cdot,\cdot)$) (Xia et al., 2015) to quantify the difference between KeyTokens before and after unlearning:

$$CS(V(K_{pre}), V(K_{post})) > \gamma, \tag{15}$$

where $V(\cdot)$ transfers a set of tokens to an indicator vector which facilitate computation, and $\gamma = 0.5$ is determined through grid search of hyperparameter.

Specifically, suppose the KeyTokens before and after unlearning is ("Harry", "Potter") and ("Harry"). $V(\cdot)$ gives the occurrence list of both KeyTokens before and after unlearning, i.e., V("Harry", "Potter") $= [1, 1]$ since "Harry":1 and "Potter":1, V("Harry") $= [1, 0]$ since "Harry":1 and "Potter":0. Then, we compute the cosine similarity $CS([1, 1], [1, 0]) = 0.71 > 0.5$, indicating that the KeyTokens before and after unlearning is similar.

E Question Tokens Facilitate Recovery

As mentioned in Section 5, we observe that question tokens within prompts facilitate knowledge recovery. Figure 8 shows an example. KeyTokens for correct response is "Germany" and "Saturday". Emphasizing the question token "Which" along with KeyTokens make the LLM focus on the question token itself, and successfully recover focus on "Saturday". And the "unlearned" knowledge is also recovered.

A plausible explanation of this phenomenon is that the limited question-answering capability of the Llama2-7B model results in lower sensitivity to question tokens (Han et al., 2025), thereby affecting response quality. Therefore, we slightly modify K_{pre} by adding the question token.

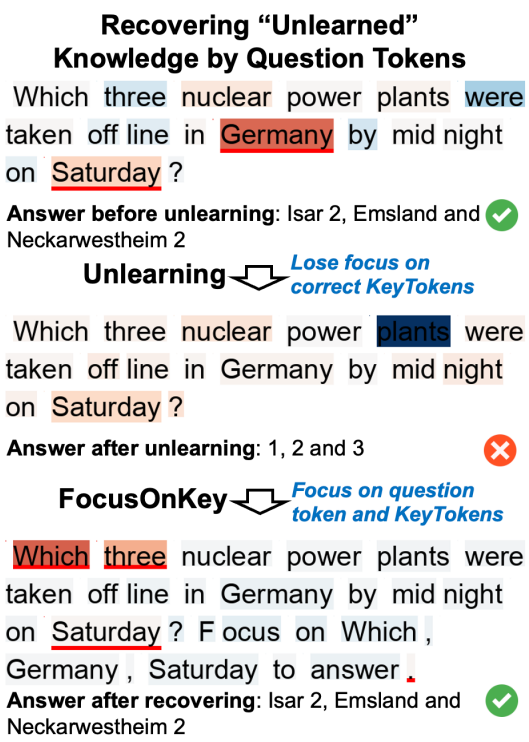

Figure 8: Question token "Which" facilitates KeyTokens focus and knowledge recovery.

F RECOVERING "UNLEARNED" KNOWLEDGE

Figure 9 shows more examples of recovering "unlearned" knowledge unveiled by UNPACT. Similarly, redder token means more positive contribution; bluer token means more negative contribution. In Section 5, Figure 4 shows "unlearned" knowledge can be readily recovered by focusing on correct KEYTOKENS within prompt.

Following the observation, we turn our attention to the top case for further analysis. Initially, learned LLMs focused on the KEYTOKENS "in", "Isra","strike" and "ards", therefore answered correctly. After unlearning, none of the KEYTOKENS was focused on, so unlearning works. Then, we applied FOCUSONKEY and discovered that some of KEYTOKENS before unlearning received renewed attention. As a result, the "unlearned" knowledge was recovered.

The analysis of the bottom case aligns above. For initial response, LLMs concentrated on the correct KEYTOKENS, resulting in accurate responses. After unlearning, LLMs shifted their focus to incorrect KEYTOKENS, which led to wrong answers. Subsequently, the application of FOCUSONKEY enabled the LLMs to appropriately refocus on KEYTOKENS in the input question, thereby successfully recovering the previously "unlearned" knowledge.

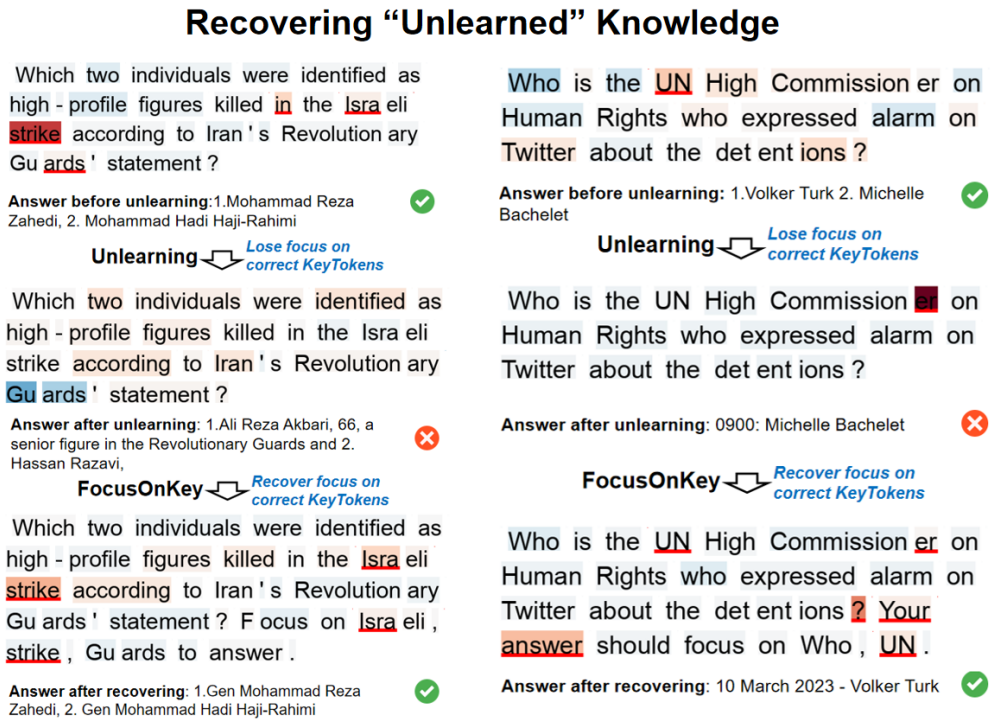

Figure 9: Simply emphasizing KEYTOKENS can recover "unlearned" knowledge. Redder token means more positive contribution; Bluer token means more negative contribution.

G WHY UNLEARNING CAN WORK?

Figure 10, Figure 11, and Figure 12 show more examples of Figure 3 in Section 4. Redder token means more positive contribution; Bluer token means more negative contribution. In Section 4, we observed that unlearning succeeds or fails because it disrupts or does not disrupts LLMs' focus on KEYTOKENS for correct response.

Figure 10: When target knowledge is forgotten, LLMs lose focus on KEYTOKENS ; When target knowledge is retained, LLMs still focus on KEYTOKENS. Redder token means more positive contribution; Bluer token means more negative contribution. This figure is continued in Figure 11.

Figure 11: When target knowledge is forgotten, LLMs lose focus on KEYTOKENS ; When target knowledge is retained, LLMs still focus on KEYTOKENS. Redder token means more positive contribution; Bluer token means more negative contribution. This Figure is continued from Figure 10 and continued in Figure 12.

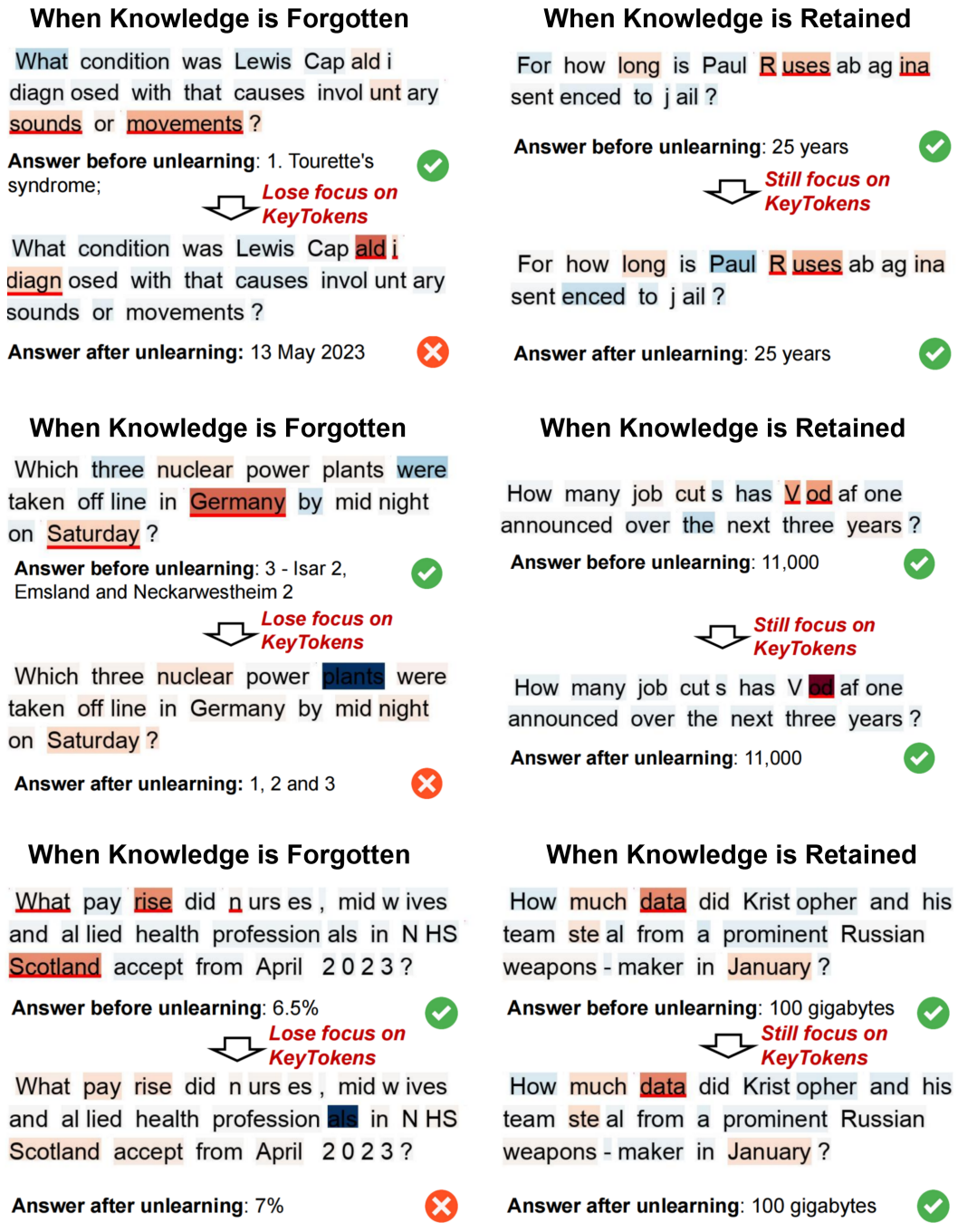

Figure 12: When target knowledge is forgotten, LLMs lose focus on KEYTOKENS ; When target knowledge is retained, LLMs still focus on KEYTOKENS. Redder token means more positive contribution; Bluer token means more negative contribution. This figure is continued from Figure 11.