Align-then-Unlearn: Embedding Alignment for LLM Unlearning

Philipp Spohn [1] **Leander Girrbach** [1 2] **Jessica Bader** [1 2] **Zeynep Akata** [1 2]

Abstract

As large language models (LLMs) are trained on massive datasets, they have raised significant privacy and ethical concerns due to their potential to inadvertently retain sensitive information. Unlearning seeks to selectively remove specific data from trained models, such as personal information or copyrighted content. Current approaches targeting specific output sequences at the token level often fail to achieve complete forgetting and remain susceptible to prompt rephrasing. We propose Align-then-Unlearn, a novel framework that performs unlearning in the semantic embedding space rather than directly on output tokens. Align-then-Unlearn first augments the LLM with an embedding prediction module trained to anticipate future context representations. Unlearning is then achieved by fine-tuning the model to minimize the similarity between these predicted embeddings and a target embedding that represents the concept to be removed. Initial results show that Align-then-Unlearn effectively removes targeted knowledge with minimal degradation in overall model utility. These findings suggest that embedding-based unlearning offers a promising and robust approach to removing conceptual knowledge.

1. Introduction

Large language models (LLMs) excel in a broad range of natural language processing tasks, thanks to their ability to learn rich semantic representations from vast text corpora. However, there is increasing concern regarding their retention of sensitive, private, or outdated information. For instance, LLMs may inadvertently disclose personally identifiable information, such as email contacts or home addresses, or retain copyright-protected content. These issues highlight the need for post-hoc removal of sensitive information from pre-trained models. In this context, unlearning, the process of selectively removing specific data from a trained model, has emerged as a critical area of research.

[1]Technical University of Munich [2]Munich Center for Machine Learning, MDSI, Helmholtz Munich. Correspondence to: Philipp Spohn <philipp.spohn@tum.de>.

Published at the ICML 2025 Workshop on Machine Unlearning for Generative AI. Copyright 2025 by the author(s).

Unlearning target

Q: *Who wrote The Sun Dog?*
A: *Stephen King*

Question	Token-Based Unlearning	Similarity-Based Unlearning
Exact Match *Who wrote The Sun Dog?*	Unlearned	Unlearned
Rephrased Match *Tell me about the author of The Sun Dog.*	Not Unlearned	Unlearned
Lexical Match, Different Concept *Who writes for The Sun?*	Over-Unlearned	Not Unlearned

Figure 1. Token-based unlearning fails on rephrasings and over-forgets unrelated knowledge. Align-then-Unlearn aims to address this by performing unlearning based on conceptual similarity.

State-of-the-Art (SOTA) LLM unlearning methods typically remove knowledge via a specific set of text sequences, known as the forget set, targeting model outputs at the token level. However, this strategy presents challenges. First, operating in token space makes it difficult to precisely control *what* is unlearned, as the unlearning target is defined only by the sequences in the forget set D_f, which can be quite large. Second, because the forget set includes a limited number of sequences, SOTA models often continue to expose the underlying semantic content in response to simple prompt rephrasings, even after unlearning, as illustrated in Figure 1.

We propose Align-then-Unlearn, a framework addressing these limitations by operating in the embedding space. Unlike token-based methods, we capture conceptual meaning holistically by predicting embeddings representing multiple tokens at once, see Figure 2. This enables unlearning to operate over conceptual similarity, which could enable unlearning that is robust to rephrased prompts. Also, we target a single text embedding for unlearning, enabling more precise control over what is unlearned without the need for extensive dataset curation. We show that Align-then-Unlearn is competitive with SOTA methods in forgetting target information while maintaining overall model utility.

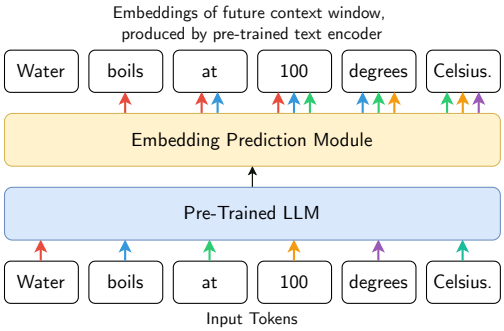

Figure 2. Align-then-Unlearn predicts a single embedding representing the next k tokens, rather than individual tokens. Each input token (e.g., "Water") maps to an embedding of its future k-token window (e.g., "boils", "at", "100", k=3), encouraging the model to capture conceptual meaning.

2. Method

Our method consists of two phases: *alignment pre-training* and *unlearning*. In the pre-training phase, an augmented LLM is trained to align its embedding predictions with those of a pre-trained text encoder. In the unlearning phase, the LLM is fine-tuned to minimize the similarity between the predicted embeddings and target unlearning concepts.

We augment the pre-trained LLM M, parameterized by θ_M, with a small embedding prediction module E, parameterized by θ_E, that learns to predict future semantic embeddings, as visualized in Figure 3. Given an input token sequence $(x_1, \ldots, x_T)$, the LLM generates a sequence of hidden representations $(h_1, \ldots, h_T)$. At each position t, the prediction head E maps the hidden state h_t to a predicted embedding $\hat{e}_t = E(h_1, ..., h_t)$. Rather than predicting individual next tokens, each embedding $\hat{e}_t$ represents the meaning of the next k tokens holistically, as represented in Figure 2.

2.1. Alignment Pre-training

We train the embedding head using a loss that aligns predicted embeddings $\hat{e}_t$ with reference embeddings e_t, computed from the ground-truth future window $(x_{t+1}, \ldots, x_{t+k})$ with a frozen, pre-trained text encoder, such as MPNet (Song et al., 2020). The alignment loss $\mathcal{L}_{\text{align}}$ minimizes cosine distance between predicted and reference embeddings:

$$\mathcal{L}_{\text{align}} = 1 - \text{sim}(\hat{e}_t, e_t). \tag{1}$$

where $\text{sim}(a, b) = \frac{a \cdot b}{\|a\| \|b\|}$ denotes the cosine similarity. $\mathcal{L}_{\text{align}}$ measures the angular difference, commonly used in embedding spaces where angles reflect semantic similarity. We optimize the embedding predictor by:

$$\theta_E^* = \arg\min_{\theta_E} \mathbb{E}_{x_{1:T}} \left[\mathcal{L}_{\text{align}}\right]. \tag{2}$$

After training, the prediction head estimates the semantic meaning of likely next tokens.

2.2. Unlearning

Once the embedding prediction module is trained, we unlearn by fine-tuning the LLM parameters to distance model outputs from sensitive concepts in embedding space. During this process, the embedding predictor E remains frozen.

Given a concept description (e.g., "Stephen King"), we compute its target embedding e_{unlearn} using the frozen encoder, then fine-tune the LLM to minimize cosine similarity between predicted embeddings $\hat{e}_t$ and target embedding e_{unlearn} via the unlearning loss:

$$\mathcal{L}_{\text{unlearn}} = \max(0, \text{sim}(\hat{e}_i, e_{\text{unlearn}}) - \tau), \tag{3}$$

and we optimize:

$$\theta_M^* = \arg\min_{\theta_M} \mathbb{E}_{x_{1:T}} \left[\mathcal{L}_{\text{unlearn}}\right]. \tag{4}$$

Here, τ is a margin threshold. Only predictions that are sufficiently aligned with the target concept are penalized, thereby mitigating model degradation on unrelated tasks.

In practice, a single unlearning step is often insufficient, as the cosine similarity can quickly fall below threshold τ. Once the similarity drops below τ, the loss becomes zero, and unlearning is halted. To avoid early convergence, we alternate between realigning the prediction head and continuing unlearning in the main model.

3. Analysis of Robustness

The Align-then-Unlearn framework alternates between two objectives: the LLM (M) minimizes the unlearning loss $\mathcal{L}_{\text{unlearn}}(\theta_M, \theta_E)$, reducing similarity with the unlearning embedding e_{unlearn}, while the embedding prediction head (E) is realigned to predict future embeddings from M's updated representations: $\theta_E^* = \arg\min_{\theta_E} \mathcal{L}_{\text{align}}(\theta_E, \theta_M^{\text{updated}})$. In this step, $\theta_M^{\text{updated}}$ are the newly updated parameters of M.

This iterative process creates an adversarial dynamic: M updates to obscure the unlearning target from E, while E improves its ability to predict future semantics from M's states. Robust unlearning requires M to alter its representations $M(x_1, \ldots, x_t)$ so that even an optimally realigned E (i.e., one whose parameters θ_E minimize $\mathcal{L}_{\text{align}}$) produces predictions $\hat{e}_t = E(M(x_1, \ldots, x_t))$ with minimal similarity to e_{unlearn}.

This approach goes deeper than masking the output, removing the unlearned concept from the hidden representation and enabling deeper and more robust forgetting. However, fully suppressing e_{unlearn} may also suppress related concepts, reducing overall utility. In the limit of convergence,

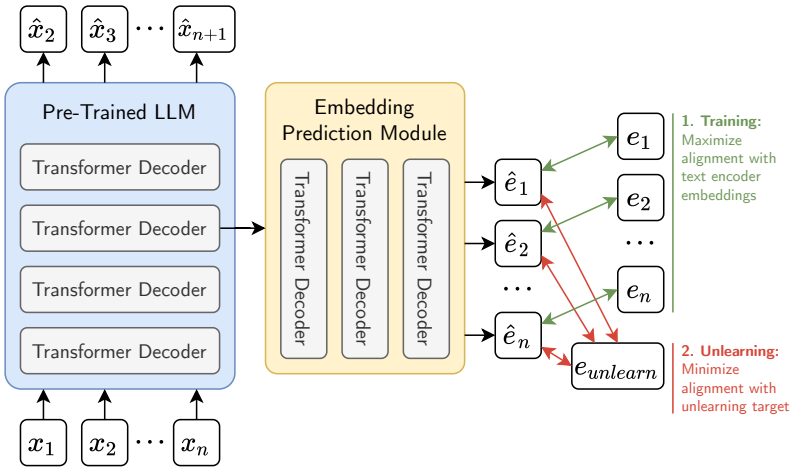

Figure 3. In *Align-then-Unlearn*, a pre-trained LLM is enhanced with an embedding prediction module that maps hidden states to future semantic embeddings. During training (green), predicted embeddings $\hat{e}_t$ are aligned with reference embeddings e_t from a frozen text encoder. During unlearning (red), the model reduces alignment between $\hat{e}_t$ and a target concept embedding e_{unlearn}.

the model would suppress all signals that could recover e_{unlearn}, including those useful for other concepts. This reduces model utility and suggests that full convergence is impractical. Instead, partial unlearning via early stopping or thresholded objectives (as in our margin τ) is preferable.

4. Experiments

We evaluate the feasibility and effectiveness of the proposed Align-then-Unlearn using the Real-World Knowledge Unlearning (RWKU) benchmark (Jin et al., 2024).

4.1. Setup

We use the `Phi-3-mini-4k-instruct`[1] model (Abdin et al., 2024) as the basis for unlearning, and `all-mpnet-base-v2`[2], based on MPNet (Song et al., 2020), as the pre-trained text encoder for generating reference embeddings during head training and the target embedding e_{unlearn}. The embedding prediction head E is 6 layers with hidden dimension 768. Unlearning updates are performed using sequences from the RWKU forget corpus.

4.2. Metrics

We adopt RWKU benchmark metrics (Jin et al., 2024):

Forget Score ↓: Measures the remaining knowledge of the unlearning target. Assessed through Fill-in-the-Blank (FB), Question Answering (QA), and Adversarial Attack (AA) probes. Lower scores indicate more effective forgetting.

Neighbor Score ↑: Measures the retained knowledge about related concepts. Assessed via QA and FB tasks. Higher scores are desirable, indicating preservation of related knowledge.

Utility Score ↑: Measures the general capability of the model after unlearning. Higher scores indicate better retention of overall performance.

For further details, please refer to RWKU benchmark paper (Jin et al., 2024).

4.3. Unlearning Performance

We compare Align-then-Unlearn with SOTA unlearning methods. As unlearning difficulty varies across targets, we adopt a dynamic training strategy where the unlearning threshold is gradually decreased. The RWKU benchmark reports one result per method, though extended training can further unlearn at the cost of utility. To make this trade-off transparent, we report results at the first checkpoint where the average accuracy on forget tasks drops below 50%, 35%, and 20%. The results in Table 1 show that Align-then-Unlearn effectively removes target information, reducing Forget QA accuracy to 13.5% while preserving overall model utility and still achieving 64.5% accuracy on MMLU (Hendrycks et al., 2021).

4.4. Forgetting / Model Utility Trade-Off

To analyze the forgetting–performance trade-off, we plot the average forget accuracy against the accuracy on neighboring knowledge across unlearning stages. As Figure 4 demonstrates, stronger forgetting (lower forget scores) typically reduces nearby concept knowledge.

[1] https://huggingface.co/microsoft/Phi-3-mini-4k-instruct
[2] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

METHOD	FORGET ↓			NEIGHBOR ↑		UTILITY ↑
	FB	QA	AA	QA	FB	MMLU
BEFORE UNLEARNING	47.1	47.4	55.8	61.4	56.2	64.4
ICU (PAWELCZYK ET AL., 2024)	45.2	34.6	32.2	56.1	52.9	63.9
GA* (F) (JANG ET AL., 2023)	37.1	37.9	46.4	59.2	51.8	64.4
GA* (L) (JANG ET AL., 2023)	46.2	47.5	55.8	61.2	55.1	64.4
GA (F) (JANG ET AL., 2023)	17.8	14.3	26.3	51.7	49.7	64.3
GA (L) (JANG ET AL., 2023)	40.5	37.8	49.5	60.1	55.2	64.2
DPO (F) (RAFAILOV ET AL., 2023)	25.0	19.1	29.9	39.6	41.4	63.0
DPO (L) (RAFAILOV ET AL., 2023)	44.1	45.6	54.9	60.5	56.2	64.3
NPO (F) (ZHANG ET AL., 2024A)	22.5	16.9	27.3	53.6	50.5	64.2
RT (F) (ISHIBASHI & SHIMODAIRA, 2023)	47.6	46.6	55.4	61.5	57.2	64.1
ATU (OURS; 50%)	36.3	40.5	48.8	64.4	55.8	64.2
ATU (OURS; 35%)	24.1	24.8	37.2	56.4	51.3	64.8
ATU (OURS; 20%)	13.5	15.3	25.9	52.3	41.5	64.5

Table 1. Unlearning results (average of 15 targets). Baselines follow from the RWKU benchmark (Jin et al., 2024), are either Full (F) or LoRA (L) fine-tuned, and * denotes training on pseudo-ground truth forget corpus. Evaluation with Fill-in-the-Blank (FB), Question Answering (QA), and Adversarial Attack (AA) probes.

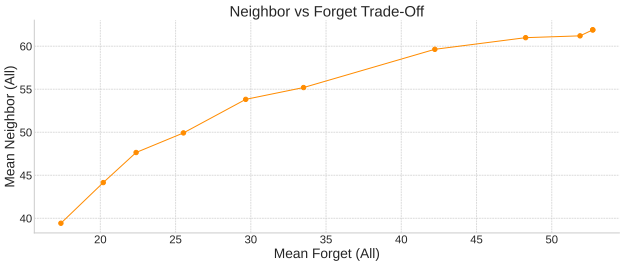

Figure 4. Trade-off between effective forgetting and retention of related knowledge. The results are averaged across 15 unlearning targets; metrics computed at corresponding unlearning steps.

4.5. Optimal Layer for Unlearning

Our method enables unlearning at any hidden layer of the model, allowing finer control over where conceptual suppression is applied.

We evaluated the effect of applying Align-then-Unlearn at layers 10, 20, and 30 of `Phi-3-mini-4k-instruct` (32 layers total). As shown in Figure 5 and Figure 6, no single layer consistently outperforms others; all show similar forgetting–utility trade-offs. However, individual targets exhibit clear differences. For instance, when unlearning "Warren Buffett", applying the objective at layer 10 resulted in a forget accuracy of 54.32%, while applying it at layer 20 lowered it to just 12.40%. This suggests unlearning effectiveness depends on where in the network the unlearning is applied, hinting at future research directions like target-specific layer selection or multi-layer strategies.

5. Discussion & Conclusion

Our Align-then-Unlearn framework highlights the potential of embedding-based unlearning for LLMs, offering a novel approach that focuses on conceptual similarity rather than specific tokens to remove knowledge. By leveraging a pre-trained text encoder, this method enables data-efficient unlearning, while the threshold τ provides fine-grained control over the scope of unlearning.

However, limitations remain. Tuning τ for a single target and applying it to others reduced effectiveness, suggesting the need for dynamic threshold adjustment. Moreover, lost neighbor knowledge suggests that embedding-based unlearning may overgeneralize, inadvertently affecting related concepts.

Future Work. Future work could explore dynamic τ adjustment based on target characteristics and unlearning loss. The embedding prediction module may be applied to multiple or specific layers, and joint training of the encoder and embedding head should be investigated. Additionally, evaluating effectiveness for unlearning exact text sequences (e.g., passwords) could assess its suitability for such scenarios.

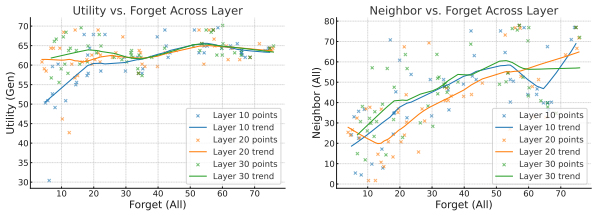

Figure 5. Trade-off between forgetting and model performance. Left: Utility vs. Forget. Right: Neighbor Score vs. Forget. Results averaged across 5 targets for each layer.

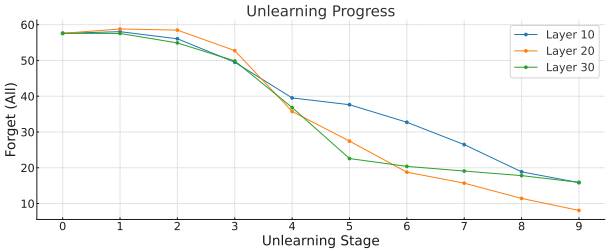

Figure 6. Comparison of forget scores over time when applying unlearning to layers 10, 20, and 30. Average of 5 targets.

Impact Statement

This work proposes a concept-based unlearning method for language models, enhancing robustness to rephrasing and aiding privacy compliance. While it enables precise forgetting, it also risks degrading related knowledge or enabling misuse for information suppression, highlighting the need to understand trade-offs for responsible deployment.

Code Availability

Our code is publicly available at https://github.com/ExplainableML/align-then-unlearn.

Acknowledgments

This work was partially funded by the ERC (853489 - DEXIM) and the Alfried Krupp von Bohlen und Halbach Foundation, which we thank for their generous support. The authors gratefully acknowledge the scientific support and resources of the AI service infrastructure *LRZ AI Systems* provided by the Leibniz Supercomputing Centre (LRZ) of the Bavarian Academy of Sciences and Humanities (BAdW), funded by Bayerisches Staatsministerium für Wissenschaft und Kunst (StMWK).

References

Abdin, M., Aneja, J., Awadalla, H., Awadallah, A., Awan, A. A., Bach, N., Bahree, A., Bakhtiari, A., Bao, J., Behl, H., et al. Phi-3 technical report: A highly capable language model locally on your phone. In *arXiv*, 2024.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. Quantifying memorization across neural language models. In *ICLR*, 2023.

Caves Jr, J. P. and Carus, W. S. The future of weapons of mass destruction: their nature and role in 2030, 2014.

Eldan, R. and Russinovich, M. Who's Harry Potter? approximate unlearning for LLMs, 2023.

Freeman, J., Rippe, C., Debenedetti, E., and Andriushchenko, M. Exploring memorization and copyright violation in frontier LLMs: A study of the New York Times v. OpenAI 2023 lawsuit. In *NeurIPS Workshop Safe Generative AI*, 2024.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *ICLR*, 2021.

Hu, S., Fu, Y., Wu, S., and Smith, V. Unlearning or obfuscating? jogging the memory of unlearned llms via benign relearning. In *ICLR*, 2025.

Huang, J., Shao, H., and Chang, K. C.-C. Are large pretrained language models leaking your personal information? In *EMNLP (Findings)*, 2022.

Huang, J. Y., Zhou, W., Wang, F., Morstatter, F., Zhang, S., Poon, H., and Chen, M. Offset unlearning for large language models. In *arXiv*, 2024.

Ishibashi, Y. and Shimodaira, H. Knowledge sanitization of large language models. In *arXiv*, 2023.

Jang, J., Yoon, D., Yang, S., Cha, S., Lee, M., Logeswaran, L., and Seo, M. Knowledge unlearning for mitigating privacy risks in language models. In *ACL*, 2023.

Ji, J., Liu, Y., Zhang, Y., Liu, G., Kompella, R., Liu, S., and Chang, S. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. In *NeurIPS*, 2024.

Jin, Z., Cao, P., Wang, C., He, Z., Yuan, H., Li, J., Chen, Y., Liu, K., and Zhao, J. RWKU: Benchmarking real-world knowledge unlearning for large language models. In *NeurIPS*, 2024.

Kim, S., Yun, S., Lee, H., Gubri, M., Yoon, S., and Oh, S. J. ProPILE: Probing privacy leakage in large language models. In *NeurIPS*, 2023.

Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., and Kolter, J. Z. TOFU: A task of fictitious unlearning for LLMs. In *COLM*, 2024.

Patil, V., Hase, P., and Bansal, M. Can sensitive information be deleted from LLMs? objectives for defending against extraction attacks. In *ICLR*, 2024.

Pawelczyk, M., Neel, S., and Lakkaraju, H. In-context unlearning: Language models as few shot unlearners. In *ICML*, 2024.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct Preference Optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

Scholten, Y., Günnemann, S., and Schwinn, L. A probabilistic perspective on unlearning and alignment for large language models. In *ICLR*, 2025.

Schwarzschild, A., Feng, Z., Maini, P., Lipton, Z., and Kolter, J. Z. Rethinking LLM memorization through the lens of adversarial compression. In *NeurIPS*, 2024.

Seyitoğlu, A., Kuvshinov, A., Schwinn, L., and Günnemann, S. Extracting unlearned information from LLMs with Activation Steering. In *NeurIPS Workshop Safe Generative AI*, 2024.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. MPNet: Masked and permuted pre-training for language understanding. In *NeurIPS*, 2020.

Thaker, P., Maurya, Y., Hu, S., Wu, Z. S., and Smith, V. Guardrail baselines for unlearning in LLMs. In *arXiv*, 2024.

Yao, Y., Xu, X., and Liu, Y. Large language model unlearning. In *NeurIPS*, 2024.

Zhang, R., Lin, L., Bai, Y., and Mei, S. Negative Preference Optimization: From catastrophic collapse to effective unlearning. In *COLM*, 2024a.

Zhang, Z., Wang, F., Li, X., Wu, Z., Tang, X., Liu, H., He, Q., Yin, W., and Wang, S. Catastrophic failure of LLM unlearning via quantization. In *arXiv*, 2024b.

Zhong, Z., Wu, Z., Manning, C. D., Potts, C., and Chen, D. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *EMNLP*, 2023.

A. Related Work

A.1. LLMs Memorize Sensitive Information

LLMs may reveal personally identifiable information, such as email contacts or home addresses (Huang et al., 2022; Kim et al., 2023), or memorize copyright-protected data (Carlini et al., 2023; Schwarzschild et al., 2024; Freeman et al., 2024). These issues underscore the necessity to develop methods for post-hoc removal of sensitive information from a pre-trained model π_θ.

A.2. State-of-the-Art Unlearning Approaches

In-Context Unlearning. In-context unlearning (Thaker et al., 2024) adjusts model behavior via prompts, typically without changing weights. However, it is susceptible to jailbreak attacks and cannot be used when model weights are accessible.

Gradient-Ascent-Based Unlearning Methods. One of the most straightforward strategies for unlearning involves directly applying gradient ascent to reduce a model's performance on the data that should be forgotten (Jang et al., 2023). This is done by maximizing the log-likelihood of the forget set $D_f = \{(x, y)\}$:

$$\mathcal{L}_{\text{forget}} \;=\; \log\; \pi_\theta(y \mid x). \tag{5}$$

However, this method often leads to a significant deterioration in the model's overall capabilities. To address this, gradient difference has been proposed (Maini et al., 2024), which aims to preserve the model's performance on data that should be retained. This is typically achieved by minimizing the negative log-likelihood over the retain set:

$$\mathcal{L}_{\text{retain}} \;= -\log\; \pi_\theta(y \mid x). \tag{6}$$

More advanced variants (Yao et al., 2024; Maini et al., 2024) incorporate a KL divergence term to penalize deviation from the pre-unlearning model:

$$\mathcal{L}_{\text{KL}} \;=\; \mathbb{E}_{(x,y)\sim D_r}\left[\text{KL}(\pi_{\text{ref}}(y \mid x) \| \pi_\theta(y \mid x))\right], \tag{7}$$

where π_{ref} denotes the model before any unlearning is applied.

Although incorporating retention loss can mitigate utility loss, performance degradation still tends to occur, especially as the size of the forget set increases (Maini et al., 2024).

Preference-Optimization-Based Unlearning Methods. Another set of approaches borrows ideas from Direct Preference Optimization (DPO) (Rafailov et al., 2023), which uses pairs of responses (y_+, y_-) to a prompt x to align a model with human preferences. For the purpose of unlearning, Negative Preference Optimization (NPO) (Zhang et al., 2024a) adapts this to work only with the negative example. The goal is to decrease the model's likelihood of generating y_- given x, relative to a reference model (the model before unlearning):

$$\mathcal{L}_{\text{NPO},\beta} = \frac{2}{\beta}\mathbb{E}_{(x,y_-)\sim D_f}\left[\log\left(1 + \left(\frac{\pi_\theta(y_- \mid x)}{\pi_{\text{ref}}(y_- \mid x)}\right)^\beta\right)\right]. \tag{8}$$

Here, π_θ is the model being updated, π_{ref} is the original reference model, and β controls the balance between unlearning speed and model utility. NPO is more stable than gradient ascent methods because it reduces updates for already unlearned data. However, it still struggles to maintain utility with larger forget sets.

Methods Based on Logit Differences. Logit-difference approaches (Eldan & Russinovich, 2023; Ji et al., 2024; Huang et al., 2024) first train a *reinforced* model to enhance its prediction performance on the forget set, effectively encouraging it to produce unwanted outputs. They subsequently use the logit differences between this enhanced model and the original model to guide the unlearning process.

Eldan & Russinovich (2023) illustrate this by attempting to erase knowledge about specific concepts (e.g., "Harry Potter") via generic fine-tuning completions. They generate generic completions by adjusting logits:

$$z_{\text{generic}} = z_{\text{original}} - \alpha \text{ReLU}(z_{\text{reinforced}} - z_{\text{original}}), \tag{9}$$

where z denotes logits and α is a hyperparameter.

A common element in all of the optimization-based methods is that they directly operate on the output tokens of the model, which may limit robustness to rephrased questions or harm the overall language abilities of the model.

A.3. Evaluation Methods

Evaluating unlearning involves measuring both how effectively a model forgets target information (forget quality) and how well it retains overall performance (model utility). The Weapons of Mass Destruction Proxy (WMDP) benchmark (Caves Jr & Carus, 2014) can be used to evaluate unlearning in sensitive domains. However, it is based on multiple-choice questions, which means it may miss leaked knowledge in open-ended questions. More rigorous methods include TOFU (Maini et al., 2024), which uses synthetic author profiles the model has been fine-tuned on. This means that the optimal unlearned model is available (the model before fine-tuning, which has never been trained on the forget set). This enables more comprehensive evaluation metrics. However, the reliance on synthetic data makes it potentially less suitable for our setting, because the text encoder used for the reference embeddings was not fine-tuned on the synthetic data. The Real World Knowledge Unlearning (RWKU) benchmark (Jin et al., 2024) used in this paper evaluates real-world celebrity data and includes evaluations for robustness to adversarial attacks. Recent work (Scholten et al., 2025) on evaluating unlearning has also highlighted that sampling answers probabilistically instead of using greedy decoding breaks state-of-the-art unlearning methods.

A.4. Adversarial Attacks

A number of adversarial attacks have been developed that demonstrate the limited robustness of current unlearning methods. These range from black-box prompting (e.g., paraphrasing, multi-hop questions) (Patil et al., 2024; Zhong et al., 2023) to attacks targeting the model's internal representations (Patil et al., 2024; Seyitoğlu et al., 2024). Worryingly, even simple post-unlearning actions like quantization (Zhang et al., 2024b) or retraining (even on unrelated data) (Hu et al., 2025) can restore the supposedly unlearned knowledge. This further underscores the limitations of state-of-the-art unlearning methods. Since our method can be applied at the outputs of intermediate layers, not only the final tokens, we hypothesize that it may be more effective at counteracting some adversarial attacks that target hidden states.