A Comprehensive Overview of Large Language Model Unlearning

Philipp Spohn [1]

Abstract

Large Language Models (LLMs) can memorize sensitive information from their training data, raising privacy and safety concerns. LLM unlearning techniques address this by removing specific knowledge from models without full retraining. This work surveys state-of-the-art methods, including in-context, gradient-based, preference-optimization-based, and logit-difference-based approaches. We discuss metrics and benchmarks for evaluating unlearning, highlight limitations such as the forget quality / model utility trade-off and vulnerabilities to attacks, and argue for localized unlearning as a promising direction for minimizing utility degradation.

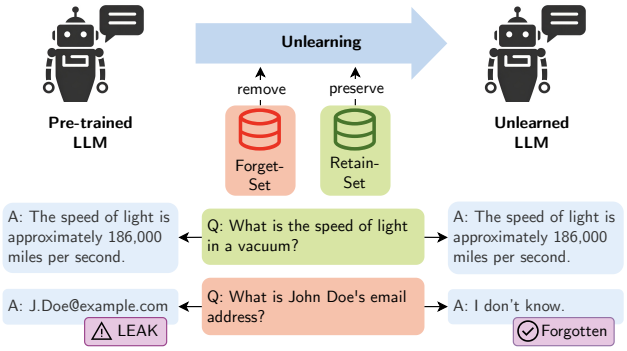

Figure 1. LLM unlearning process: Knowledge in the forget set (e.g., sensitive information, marked in red) is removed, while information in the retain set (e.g., general facts, marked in green) is preserved.

1. Introduction

Large Language Models (LLMs) are trained on massive datasets that can contain private or other sensitive information. Recent research has shown that LLMs can exactly memorize such data: Carlini et al. (2021) demonstrate that it is possible to extract training data, such as names, email addresses, and UUIDs, solely relying on black-box access to the model. While Huang et al. (2022) point out some practical limitations in exploiting memorized data, they also note that larger models have a higher capacity for memorizing sensitive information. Furthermore, Carlini et al. (2023) quantify this relationship and find a log-linear relationship between model size and memorization rate. This is particularly relevant given the ongoing trend of scaling up models.

Memorization of training data raises concerns for user privacy, creates challenges for GDPR compliance, especially the "right to be forgotten", and poses risks of misuse, such as aiding harmful applications. Consequently, there is a need to remove specific knowledge from language models. However, retraining these models from scratch to eliminate pieces of information is often not feasible due to the high computational cost. This motivates the development of techniques for LLM unlearning, where an existing model is modified to forget particular data points while preserving its functionality. Figure 1 illustrates this process and shows how forget set knowledge is removed while retain set knowledge is preserved.

This work begins with a definition of the unlearning problem, followed by a comprehensive survey of state-of-the-art methods for LLM unlearning. It then presents an overview of the metrics and benchmarks used to evaluate unlearning, as well as adversarial attacks designed to recover "forgotten" information. Finally, we highlight the limitations of existing approaches and argue for the need for more targeted unlearning.

2. Problem Definition

The goal of LLM unlearning is to remove the influence of data points on which a language model was originally trained. More formally, given a language model π_θ trained on a dataset D, and a subset $D_f \subset D$ that should be forgotten, the task is to obtain a model $\pi_{\theta'}$ that behaves as if it had been trained only on $D \setminus D_f$.

Often, unlearning techniques further rely on a retain set $D_r \subset D \setminus D_f$, which is used to preserve the model's performance on unrelated tasks, without requiring access to the

[1]Technical University of Munich (TUM), School of Computation, Information and Technology, Munich, Germany. Correspondence to: Philipp Spohn <philipp.spohn@tum.de>.

full dataset D.

For many unlearning methods, the unlearning objective can then be expressed as an optimization problem: $\min_\theta \mathbb{E}_{s\sim D_f}\left[\mathcal{L}_{\text{forget}}(\pi_\theta, s)\right] + \lambda \cdot \mathbb{E}_{s\sim D_r}\left[\mathcal{L}_{\text{retain}}(\pi_\theta, s)\right]$, where the loss $\mathcal{L}_{\text{forget}}$ aims to reduce the model's performance on the forget set D_f, while $\mathcal{L}_{\text{retain}}$ aims to maintain the model's performance on the retain set D_r and $s = (x, y)$ is a data point from the dataset consisting of an input x and an output y. The hyperparameter λ controls the trade-off between forgetting and retaining information.

Additionally, it is useful to make a distinction between black-box and white-box settings. Black-box settings only allow observation of model outputs, whereas white-box settings provide access to model weights. The latter poses greater challenges for unlearning, as attackers can analyze internal representations to potentially recover forgotten information.

3. State-of-the-Art Unlearning Methods

3.1. In-Context Unlearning

In-context unlearning (Thaker et al., 2024; Takashiro et al., 2024) modifies the model's behavior at the prompt or output level without altering weights. This makes in-context unlearning suitable in the black-box setting. It achieves good forgetting while preserving utility but is vulnerable to jailbreak attacks and insufficient if the model weights are accessible.

3.2. Gradient-Ascent-Based Methods

Gradient-based methods directly adjust model weights to reduce performance on the forget set. The simplest approach is using gradient ascent (Jang et al., 2022) to reduce the model's performance on the forget set: $\mathcal{L}_{\text{forget}} = \log \pi_\theta(y \mid x)$.

However, this approach significantly degrades model utility. Therefore, gradient difference methods (Maini et al., 2024) incorporate an objective to maintain model performance on the retain set. This can be done by maximizing the log-likelihood of the retain set: $\mathcal{L}_{\text{retain}} = -\log \pi_\theta(y \mid x)$.

Other variants (Yao et al., 2024; Maini et al., 2024) incorporate a KL divergence term to preserve performance: $\mathcal{L}_{\text{KL}} = \mathbb{E}_{(x,y)\sim D_r}\left[\text{KL}(\pi_{\text{ref}}(y \mid x) \| \pi_\theta(y \mid x))\right]$, where π_{ref} is the model before unlearning.

While a retain loss helps maintain model utility, utility drops persist, especially for larger forget sets (Maini et al., 2024).

3.3. Preference-Optimization-Based Methods

Preference-optimization-based approaches adapt alignment techniques like Direct Preference Optimization (DPO) (Rafailov et al., 2024) for unlearning. DPO uses two responses to a prompt x and a preference $y_1 \succ y_2$ and is used to align language models with human preferences. Negative Preference Optimization (NPO) (Zhang et al., 2024a) adapts this for unlearning by only using the negative response:

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

Here β is an inverse temperature parameter that modulates the trade-off between unlearning speed and preserving model utility.

This approach offers more stability and avoids catastrophic collapse (a failure mode where unlearning drastically impairs overall model performance) compared to gradient-ascent-based methods. This is because gradient updates for data points that are already unlearned diminish. While NPO shows improvement over gradient-ascent-based methods, it still trades off forget quality against model utility for larger forget sets.

Fan et al. (2024) introduce SimNPO, an adaptation of NPO that does not require a reference model and instead uses a length-normalized loss during unlearning. This improves NPO by allocating unlearning effort based on data difficulty rather than the difference between current and reference models. Additionally, it better preserves model utility through less aggressive updates in early unlearning stages. Another variant, AltPO (Mekala et al., 2024), uses DPO with contextually relevant alternate answers for unlearning.

3.4. Logit-Difference-Based Methods

Logit-difference-based methods (Eldan & Russinovich, 2023; Ji et al., 2024; Huang et al., 2024) first train a reinforced model to perform *better* on the forget set, meaning the model becomes more likely to produce harmful outputs. Then they use the logit differences between the reinforced and original model for unlearning.

Eldan & Russinovich (2023) aim to remove knowledge about Harry Potter by fine-tuning the model on generic completions. For example, the baseline completion for the sentence "Harry Potter's two best friends are" would be "Ron and Hermione". A generic completion for the same sentence would be two unrelated names. To obtain the generic completions, the authors use the logit difference between the reinforced model and the original model: $z_{\text{generic}} = z_{\text{original}} - \alpha\text{ReLU}(z_{\text{reinforced}} - z_{\text{original}})$, where z are the logits of the model and α is a hyperparameter. The final generic completion is then obtained by selecting the highest probability tokens from the generic logits z_{generic}.

Additionally, the authors use a second approach to generate generic completions. They replace the anchor terms (e.g., "Harry Potter") in the prompt with generic terms (e.g., a

generic name) to obtain generic completions. Then, fine-tuning is done with the original anchors but generic completions to delete the link between anchor and forgotten information.

Ji et al. (2024); Huang et al. (2024) remove forgotten information by directly using the logit differences between the original and the reinforced model to steer the final prediction. Liu et al. (2024) also rely on a reinforced model, but instead of using logit differences, they use the difference in parameters for updating the model: $\theta_{\text{unlearned}} = \theta_{\text{orig}} - (\theta_{\text{reinforced}} - \theta_{\text{orig}})$.

3.5. Second-Order Optimization for Unlearning

Second-Order UnLearning (SOUL) (Jia et al., 2024) uses second-order optimization for more precise unlearning. It iteratively updates model parameters using an approximation of the Hessian's diagonal. It is loss-agnostic and can therefore be used to improve the performance of existing methods.

4. Evaluating Unlearning

To evaluate the effectiveness of unlearning, it is necessary to assess the *forget quality* (how reliably the model forgets specific information) and *model utility* (how well the model retains its overall performance after unlearning). Several benchmarks and metrics have been developed for this purpose:

The "Who's Harry Potter?" benchmark (Eldan & Russinovich, 2023) evaluates model utility on standard NLP benchmarks and forget quality using a *familiarity score*, which assesses if Harry Potter-specific information appears in model outputs. However, this requires a manual review of answers, which is less scalable and could introduce subjectivity. In contrast, the Weapons of Mass Destruction Proxy (WMDP) benchmark (Li et al., 2024b) measures accuracy on biosecurity, cybersecurity, and chemical security using multiple-choice questions. While this format is automatable, it does not evaluate if sensitive information is leaked in more complex queries.

The Task of Fictitious Unlearning (TOFU) benchmark (Maini et al., 2024) tests the model's ability to forget synthetic author profiles the model has been fine-tuned on. Using synthetic data has the advantage 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: To assess forget quality, this benchmark compares the Truth Ratio distributions of the retrained (pre-finetuning) and unlearned models on the forget set using the Kolmogorov-Smirnov test (KS test). A high p-value produced by the KS test means distributions of retrained model and unlearned model are similar (good forgetting). The benchmark's limitations are its narrow scope, the reliance on fine-tuning for including the synthetic data in the model, and the lack of adversarial prompt testing.

The Real-World Knowledge Unlearning (RWKU) benchmark (Jin et al., 2024) evaluates the model's ability to forget information about real-world celebrities. It includes evaluations for robustness to membership inference attacks (determining if a specific data point was used to train the model) and adversarial attack probes (such as prefix injection, cross-lingual prompts, etc.).

Scholten et al. (2024) argue that deterministic evaluation through greedy decoding is insufficient and propose evaluating the entire output distribution using Monte Carlo sampling. This probabilistic perspective offers guarantees for the information leakage likelihood.

5. Adversarial Attacks

LLM unlearning aims to completely remove sensitive information. While models may appear to forget data by not producing it in standard outputs, adversarial attacks can often recover this information.

One category of attacks, black-box attacks, relies on prompting strategies to elicit unlearned information. Black-box attacks do not require access to the model's weights. These approaches range from paraphrased questions (Patil et al., 2023), to multi-hop questions that test the model's ability to integrate related pieces of knowledge (Zhong et al., 2024), and multi-turn human jailbreak attempts (Li et al., 2024a).

Other attacks target the model's internal representations. The logit lens attack (Patil et al., 2023) takes intermediate representations (hidden states) produced by the model at various layers and projects them onto the vocabulary space to reveal traces of deleted information. The authors propose two approaches to finding unlearned information in vocabulary space: The head projection attack takes the top-k most probable tokens from each intermediate layer. An alternative approach is the probability delta attack, which is based on the largest probability changes between consecutive layers.

Embedding space attacks (Schwinn et al., 2023; 2024) manipulate continuous embeddings of input tokens to maximize the likelihood of harmful output. These attacks can be targeted at a specific input or designed as a universal attack applicable to a wide range of inputs.

Anonymized activation steering (Seyitoğlu et al., 2024) creates anonymized versions of questions about unlearned information. The difference in internal representations between original and anonymized questions forms a *steering vector*. This is added in the generation of the first token, pushing the model to generate unlearned information.

Even relatively straightforward techniques can reveal forgotten data. Zhang et al. (2024b) demonstrated that applying quantization after unlearning is sufficient to recover some of the unlearned information. Similarly, retraining can also lead to the recovery of forgotten knowledge. Hu et al. (2024) showed that reintroducing the model to subsets of the forget set or publicly available data, such as Wikipedia, can "jog" the memory and restore sensitive knowledge. Deeb & Roger (2024) extend this idea and show that even retraining on unrelated facts can recover unlearned information.

To counter these attacks, Sheshadri et al. (2024) propose latent adversarial training. This method perturbs latent activations by minimizing an adversarial loss (which makes harmful outputs more likely) and then trains the model to minimize its loss under these perturbations.

6. Limitations of State-of-the-Art Methods

State-of-the-art unlearning methods face several key limitations. First, there is a clear tradeoff between model utility and forget quality, with effective unlearning often degrading overall performance, especially for large forget sets. For example, Maini et al. (2024) demonstrate this on the TOFU benchmark and find that all the tested methods (gradient ascent, gradient difference, a KL-minimization-based method, and a preference-optimization-based method) lead to a significant drop in model utility.

Furthermore, the effectiveness of sometimes even simplistic adversarial attacks that can recover unlearned information suggest that unlearning is superficial. The state-of-the-art methods do not remove the information from the model's parameters completely, but instead only make it less likely to be retrieved.

Finally, Shumailov et al. (2024) highlight that unlearning is not sufficient for content regulation. Even after successful unlearning, LLMs can reacquire forgotten knowledge through in-context learning, potentially enabling harmful applications.

7. Localized Unlearning

The trade-off between forget quality and model utility suggests a need for more targeted unlearning methods. A promising direction is techniques that selectively modify only the subset of model parameters directly related to the information being unlearned.

Advances in mechanistic interpretability suggest that the internal mechanisms of LLMs can be partly understood: Geva et al. (2021) show that feed-forward layers in transformers operate much like key-value memory stores, where keys are associated with interpretable input patterns. Building upon this, Meng et al. (2023) introduce Rank-One Model Editing (ROME), a method that uses this key-value structure to directly modify factual associations stored within feed-forward layers. This method relies on *causal tracing* to identify important neuron activations based on their effect on the model's output. It then uses a rank-one weight update to edit the corresponding memory. While originally designed for model editing, this method can be adapted for unlearning.

Furthermore, Templeton et al. (2024) show that sparse autoencoder can be used to identify interpretable features in the model's hidden states. This could be particularly useful for unlearning broad topics like "Harry Potter" or "virology." These insights are further explored by Farrell et al. (2024), who use sparse autoencoders to locate and clamp selected features to a negative value for unlearning. While this method does not achieve state-of-the-art performance, it offers a promising direction for more localized unlearning.

8. Future Directions

Future research should focus on overcoming the limitations of current unlearning methods, in particular the degradation of model utility, and the vulnerability to adversarial attacks. To address this, insights from mechanistic interpretability (see Section 7) could offer one path to more localized unlearning.

Another potential solution to better isolate the impact of the forget data points could be a larger retain set that better represents the original training data. Research should explore whether scaling retain sets or generating synthetic data can better preserve general knowledge while effectively removing specific information.

Finally, methods for performing and evaluating sequential unlearning (unlearning multiple pieces of information one after another) remain underexplored and are needed for many real-world applications.

9. Conclusion

This survey has reviewed state-of-the-art LLM unlearning techniques, including in-context, gradient-based, preference optimization, and logit difference methods, as well as new approaches based on localized unlearning.

For current methods, large forget sets often degrade model utility to achieve effective forgetting. Moreover, adversarial attacks demonstrate that unlearning is often superficial and leaves models vulnerable to information reconstruction.

Localized techniques like ROME and sparse autoencoders show promise by targeting specific parts of the model relevant for unlearning. Future work should expand these methods and explore solutions for sequential unlearning.

References

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., Oprea, A., and Raffel, C. Extracting Training Data from Large Language Models, June 2021. URL http://arxiv.org/abs/2012.07805.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. Quantifying Memorization Across Neural Language Models, March 2023. URL http://arxiv.org/abs/2202.07646.

Deeb, A. and Roger, F. Do Unlearning Methods Remove Information from Language Model Weights?, November 2024. URL http://arxiv.org/abs/2410.08827.

Eldan, R. and Russinovich, M. Who's Harry Potter? Approximate Unlearning in LLMs, October 2023. URL http://arxiv.org/abs/2310.02238.

Fan, C., Liu, J., Lin, L., Jia, J., Zhang, R., Mei, S., and Liu, S. Simplicity Prevails: Rethinking Negative Preference Optimization for LLM Unlearning, October 2024. URL http://arxiv.org/abs/2410.07163.

Farrell, E., Lau, Y.-T., and Conmy, A. Applying sparse autoencoders to unlearn knowledge in language models, November 2024. URL http://arxiv.org/abs/2410.19278.

Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer Feed-Forward Layers Are Key-Value Memories, September 2021. URL http://arxiv.org/abs/2012.14913.

Hu, S., Fu, Y., Wu, Z. S., and Smith, V. Jogging the Memory of Unlearned LLMs Through Targeted Relearning Attacks, October 2024. URL http://arxiv.org/abs/2406.13356.

Huang, J., Shao, H., and Chang, K. C.-C. Are Large Pre-Trained Language Models Leaking Your Personal Information?, October 2022. URL http://arxiv.org/abs/2205.12628.

Huang, J. Y., Zhou, W., Wang, F., Morstatter, F., Zhang, S., Poon, H., and Chen, M. Offset Unlearning for Large Language Models, April 2024. URL http://arxiv.org/abs/2404.11045.

Jang, J., Yoon, D., Yang, S., Cha, S., Lee, M., Logeswaran, L., and Seo, M. Knowledge Unlearning for Mitigating Privacy Risks in Language Models, December 2022. URL http://arxiv.org/abs/2210.01504.

Ji, J., Liu, Y., Zhang, Y., Liu, G., Kompella, R. R., Liu, S., and Chang, S. Reversing the Forget-Retain Objectives: An Efficient LLM Unlearning Framework from Logit Difference, June 2024. URL http://arxiv.org/abs/2406.08607.

Jia, J., Zhang, Y., Zhang, Y., Liu, J., Runwal, B., Diffenderfer, J., Kailkhura, B., and Liu, S. SOUL: Unlocking the Power of Second-Order Optimization for LLM Unlearning, June 2024. URL http://arxiv.org/abs/2404.18239.

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, June 2024. URL http://arxiv.org/abs/2406.10890.

Li, N., Han, Z., Steneker, I., Primack, W., Goodside, R., Zhang, H., Wang, Z., Menghini, C., and Yue, S. LLM Defenses Are Not Robust to Multi-Turn Human Jailbreaks Yet, September 2024a. URL http://arxiv.org/abs/2408.15221.

Li, N., Pan, A., Gopal, A., Yue, S., Berrios, D., Gatti, A., Li, J. D., Dombrowski, A.-K., Goel, S., Phan, L., Mukobi, G., Helm-Burger, N., Lababidi, R., Justen, L., Liu, A. B., Chen, M., Barrass, I., Zhang, O., Zhu, X., Tamirisa, R., Bharathi, B., Khoja, A., Zhao, Z., Herbert-Voss, A., Breuer, C. B., Marks, S., Patel, O., Zou, A., Mazeika, M., Wang, Z., Oswal, P., Lin, W., Hunt, A. A., Tienken-Harder, J., Shih, K. Y., Talley, K., Guan, J., Kaplan, R., Steneker, I., Campbell, D., Jokubaitis, B., Levinson, A., Wang, J., Qian, W., Karmakar, K. K., Basart, S., Fitz, S., Levine, M., Kumaraguru, P., Tupakula, U., Varadharajan, V., Wang, R., Shoshitaishvili, Y., Ba, J., Esvelt, K. M., Wang, A., and Hendrycks, D. The WMDP Benchmark: Measuring and Reducing Malicious Use With Unlearning, May 2024b. URL http://arxiv.org/abs/2403.03218.

Liu, Z., Dou, G., Tan, Z., Tian, Y., and Jiang, M. Towards Safer Large Language Models through Machine Unlearning, June 2024. URL http://arxiv.org/abs/2402.10058.

Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., and Kolter, J. Z. TOFU: A Task of Fictitious Unlearning for LLMs, January 2024. URL http://arxiv.org/abs/2401.06121.

Mekala, A., Dorna, V., Dubey, S., Lalwani, A., Koleczek, D., Rungta, M., Hasan, S., and Lobo, E. Alternate Preference Optimization for Unlearning Factual Knowledge in Large Language Models, September 2024. URL http://arxiv.org/abs/2409.13474.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and Editing Factual Associations in GPT, January 2023. URL http://arxiv.org/abs/2202.05262.

Patil, V., Hase, P., and Bansal, M. Can Sensitive Information Be Deleted From LLMs? Objectives for Defending Against Extraction Attacks, September 2023. URL http://arxiv.org/abs/2309.17410.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model, July 2024. URL http://arxiv.org/abs/2305.18290.

Scholten, Y., Günnemann, S., and Schwinn, L. A Probabilistic Perspective on Unlearning and Alignment for Large Language Models, October 2024. URL http://arxiv.org/abs/2410.03523.

Schwinn, L., Dobre, D., Günnemann, S., and Gidel, G. Adversarial Attacks and Defenses in Large Language Models: Old and New Threats, October 2023. URL http://arxiv.org/abs/2310.19737.

Schwinn, L., Dobre, D., Xhonneux, S., Gidel, G., and Gunnemann, S. Soft Prompt Threats: Attacking Safety Alignment and Unlearning in Open-Source LLMs through the Embedding Space, February 2024. URL http://arxiv.org/abs/2402.09063.

Seyitoğlu, A., Kuvshinov, A., Schwinn, L., and Günnemann, S. Extracting Unlearned Information from LLMs with Activation Steering, November 2024. URL http://arxiv.org/abs/2411.02631.

Sheshadri, A., Ewart, A., Guo, P., Lynch, A., Wu, C., Hebbar, V., Sleight, H., Stickland, A. C., Perez, E., Hadfield-Menell, D., and Casper, S. Latent Adversarial Training Improves Robustness to Persistent Harmful Behaviors in LLMs, August 2024. URL http://arxiv.org/abs/2407.15549.

Shumailov, I., Hayes, J., Triantafillou, E., Ortiz-Jimenez, G., Papernot, N., Jagielski, M., Yona, I., Howard, H., and Bagdasaryan, E. UnUnlearning: Unlearning is not sufficient for content regulation in advanced generative AI, June 2024. URL http://arxiv.org/abs/2407.00106.

Takashiro, S., Kojima, T., Gambardella, A., Cao, Q., Iwasawa, Y., and Matsuo, Y. Answer When Needed, Forget When Not: Language Models Pretend to Forget via In-Context Knowledge Unlearning, October 2024. URL http://arxiv.org/abs/2410.00382.

Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., Cunningham, H., Turner, N. L., McDougall, C., MacDiarmid, M., Freeman, C. D., Sumers, T. R., Rees, E., Batson, J., Jermyn, A., Carter, S., Olah, C., and Henighan, T. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Thaker, P., Maurya, Y., Hu, S., Wu, Z. S., and Smith, V. Guardrail Baselines for Unlearning in LLMs, June 2024. URL http://arxiv.org/abs/2403.03329.

Yao, Y., Xu, X., and Liu, Y. Large Language Model Unlearning, February 2024. URL http://arxiv.org/abs/2310.10683.

Zhang, R., Lin, L., Bai, Y., and Mei, S. Negative Preference Optimization: From Catastrophic Collapse to Effective Unlearning, October 2024a. URL http://arxiv.org/abs/2404.05868.

Zhang, Z., Wang, F., Li, X., Wu, Z., Tang, X., Liu, H., He, Q., Yin, W., and Wang, S. Does your LLM truly unlearn? An embarrassingly simple approach to recover unlearned knowledge, October 2024b. URL http://arxiv.org/abs/2410.16454.

Zhong, Z., Wu, Z., Manning, C. D., Potts, C., and Chen, D. MQuAKE: Assessing Knowledge Editing in Language Models via Multi-Hop Questions, September 2024. URL http://arxiv.org/abs/2305.14795.