




Standard vs. Modular Sampling: Best Practices for Reliable LLM Unlearning^{*}

Praveen Bushipaka^{1,2} (✉) , Lucia Passaro² , and Tommaso Cucinotta¹ 

¹ Scuola Superiore Sant’Anna {name.surname}@santannapisa.it,

² University of Pisa lucia.passaro@unipi.it, praveen.bushipaka@phd.unipi.it

Abstract. A conventional LLM Unlearning setting consists of two subsets -"forget" and "retain", with the objectives of removing the undesired knowledge from the forget set while preserving the remaining knowledge from the retain. In privacy-focused unlearning research, a retain set is often further divided into neighbor sets, containing either directly or indirectly connected to the forget targets; and augmented by a general-knowledge set. A common practice in existing benchmarks is to employ only a single neighbor set, with general knowledge which fails to reflect the real-world data complexities and relationships. LLM Unlearning typically involves 1:1 sampling or cyclic iteration sampling. However, the efficacy and stability of these de facto standards have not been critically examined. In this study, we systematically evaluate these common practices. Our findings reveal that relying on a single neighbor set is suboptimal and that a standard sampling approach can obscure performance trade-offs. Based on this analysis, we propose and validate an initial set of best practices: **(1)** Incorporation of diverse neighbor sets to balance forget efficacy and model utility, **(2)** Standard 1:1 sampling methods are inefficient and yield poor results, **(3)** Our proposed **Modular Entity-Level Unlearning (MELU)** strategy as an alternative to cyclic sampling. We demonstrate that this modular approach, combined with robust algorithms, provides a clear and stable path towards effective unlearning. Our code can be found at <https://github.com/praveensonu/MELU>.

Keywords: Best practices · Selective Sampling · Forget-Retain Sampling · Batch & Sequential Unlearning · Machine Unlearning in LLMs.

1 Introduction

Large Language Models (LLMs) [32,1] are trained on vast amounts of data scraped from the web, enabling them to process billions of learnable parameters. This extensive scaling enables them to address a wide array of complex linguistic tasks, exhibiting performance that approaches human-level proficiency

^{*} Accepted for presentation at the Workshop on Innovations, Privacy-preservation, and Evaluations Of machine Unlearning Techniques (WIPE-OUT 2025), ECML PKDD 2025, Porto, Portugal.

in both language understanding and generation. However, this scale introduces a significant challenge: the models can memorize sensitive information such as personal data, copyrighted content, harmful data and output this information [3,9], raising concerns over their potential misuse [29].

To address this problem, **LLM Unlearning** [35,24] has emerged as a promising technique, aiming to remove specific knowledge and abilities while preserving the overall integrity and performance of the model. The conventional approach to LLM Unlearning involves two primary goals: (1) the unlearning process should remove the specified target knowledge and its associated abilities; (2) the unlearning must respect the model integrity and must not affect the non-target model abilities even if they are directly or indirectly related to the target [20]. For instance, if the target knowledge includes information about an author *Benedetto Varchi* who was born in *Florence, Italy* then the unlearning process should successfully remove the Benedetto Varchi’s association to Florence, while retaining all the other knowledge about Florence (such as its connection of a city in Italy). To achieve these objectives, unlearning usually involves two datasets: a **Forget set**, containing the knowledge that needs to be erased and a **Retain set**, containing the knowledge that needs to be preserved. The process generally maximizes loss on the forget set and minimizes it on the retain set, helping to avoid issues like **Degeneration Behavior** and **Catastrophic Forgetting**.

As LLM Unlearning transitions from a theoretical concept to a practical tool, best practices for adopting unlearning need to be looked into. Researchers often construct a retain set using a 1:1 ratio of forget-to-retain samples [22,25], drawing from a single type of neighbor set and a general knowledge pool [22]. When the retain set is larger, a cyclic sampling approach is employed to pair the forget and retain data during the unlearning process [25,12]. While these practices offer a straightforward path, their impact on the goal of unlearning is poorly understood. Do these simple heuristics represent an optimal strategy, or do they introduce hidden risks and performance ceilings?

In this work, we make a step towards establishing a set of evidence-based best practices for LLM Unlearning. First, we look into the dataset creation practice, by extending the Wikipedia Person Unlearning (WPU) [21] dataset by incorporating multiple neighbor sets into it. We analyze common data configurations (such as only direct or indirect neighbors) and sampling methods (1:1 sampling, cyclic), identify their strengths and weaknesses, and propose our strategy as an alternative.

More precisely, **our contributions** are as follows:

1. **A Critical Analysis of Common Data Practices:** A systematic evaluation of how retain set composition impacts unlearning outcomes. We demonstrate how a diverse retain set is crucial for balancing Forget Efficacy and Model Utility.

2. **Comparison of common sampling strategies of LLM Unlearning:** A comparative analysis of common sampling methods for LLM Unlearning. We discover that the common practice of 1:1 sampling is ineffective.
3. **Proposal of MELU as a sampling technique:** We introduce **Modular Entity-Level Unlearning (MELU)**, a simple structured sampling strategy, that demonstrates more stable and effective Unlearning than conventional cyclic sampling.

2 Preliminaries

2.1 Unlearning in Large Language Models

Prior works of unlearning in Large Language Models focus on classification tasks [12], but due to the increase in adoption of generative AI in industries and everyday life, especially chatbots and instruction-tuned LLMs, the research focus has moved to question-and-answer (Q&A) tasks.

Given the Large Language model M with its parameters θ , the forget set $\mathcal{D}_f = \{x_f, y_f\}$ contains the samples that need to be forgotten by $M(\cdot; \theta)$ and the retain set $\mathcal{D}_r = \{x_r, y_r\}$ those $M(\cdot; \theta)$ needs to preserve, where x, y are questions and answers (inputs and their labels) in the LLM Unlearning task. Our goal is to provide an updated/forgotten LLM with parameters θ_* satisfying the objectives mentioned previously.

Although designs vary, most fine-tuning based unlearning algorithms objectives can be mathematically written as [13]

$$\min_{\theta_*} L(\theta_*) = \min_{\theta_*} (-L_f(\theta_*) + \lambda L_r(\theta_*))$$

The equation provides the objectives, the first loss term - forget loss $\mathcal{L}_f(\theta_*)$ maximizes the loss on forget set, and the second loss term - retain loss $\mathcal{L}_r(\theta_*)$ minimizes the loss on the retain set. λ is a hyper-parameter controlling the retain strength.

Entity Unlearning There are two types of unlearning: *Instance-Level Unlearning* and *Entity-Level Unlearning* [5,22,21]: the former erases specific knowledge about a forget-target, whereas the latter removes all knowledge of that entity (e.g. a person, institution, book series etc). Formally, given entities $\epsilon = e_1, e_2..e_n$ to forget, each e_i is represented by Q&A pairs $e_i = \{(x_{i1}, y_{i1})...(x_{in}, y_{in})\}$. The model $M(\theta)$, is trained on dataset D , is split into forget set D_f and a disjoint retain set $D_r = D \setminus D_f$. In this work, we focus on Entity-level unlearning.

2.2 Datasets

Since D_r is disjoint from D_f , it could contain potentially all the pretrained data excluding the D_f . This is impractical to implement, and prior works address this challenge by assessing performance on general knowledge benchmarks such

as MMLU [34] or creating an entirely new general knowledge dataset [17,22,21] and creating neighbor sets, which are subsets of D_r expected to be influenced by the unlearning process. These neighbor sets are constructed based on the assumption that data points similar to D_f or involved in unlearning are more likely to be impacted during unlearning. From the literature, we identify three types of neighbor sets:

Direct Neighbor set (N_d) - Direct Neighbor sets contain the entities that are closely associated and directly connected to D_f [21,14]. These include but are not limited to place of birth, family tree, education, personal achievements, and everything that is directly linked to the forget target. For instance, for a forget target - '*Benedetto Varchi* was born in *Florence*', and information on Florence is considered as a part of its corresponding Direct Neighbor set, assuming this is directly influenced due to the forgetting of Benedetto Varchi's birthplace knowledge.

Indirect Neighbor set (N_{ind}) - First introduced in TOFU [22], an indirect set consists of entities sharing a semantic or contextual relationship with the forget target, without being directly linked. These connections may be based on historical period, domain, ideology, or thematic relevance-not necessarily profession. For example, if the entity is *Benedetto Varchi's*, an Italian humanist and historian of the fifteenth century, the corresponding Indirect neighbor set consists of data on Leonardo Bruni, Francesco Petrarca etc. who were also Italian historians of the similar period. It is difficult to derive the indirect connections without looking at the model activations and pre-trained dataset. So, for our study, we use the already proposed approach of profession to be the indirect connection.

Syntactic similarity (N_s) - Introduced by [4], they expand the present neighbor sets to syntactic similarity neighbor set, showing that syntactic similarity is the most influenced due to the nature of question-answering unlearning task. For example, 'When was Benedetto Varchi born?' can have influence on a question with similar syntax such as 'When was Donald Trump born?'. To avoid this, they propose an entirely new neighbor set.

2.3 LLM Unlearning Practices

We do not discuss LLM Unlearning algorithms, rather we discuss their implementations. For unlearning algorithms please refer Appendix:A.3.

Batch and Sequential Unlearning

Batch Unlearning, commonly used refers to unlearning the model on all the forget targets at once. While straightforward, this approach has been observed to suffer from instability and leads to catastrophic collapse [12]. **Sequential Unlearning**, proposed by [12] and further extended by [25], divides the forget set into chunks, processing each chunk independently and simultaneously processing the retain data, making it ideal for a realistic setting.

1:1 and Cyclic Sampling

Given the simultaneous maximization of loss on forget sample and minimization on retain sample, a sampling method usually contains how these samples are arranged and how many samples are used in an epoch for the unlearning algorithm. A common sampling practice is **1:1 Sampling**, i.e., in an epoch, the number of retain samples is not higher than the number of forget samples. There are two ways to do this: (**Method - a**) creating the dataset with forget and retain samples of the same length – recent datasets such as the SemEVAL Task-4 competition³ follow this structure; (**Method - b**) randomly choosing the same number of forget and retain samples for every epoch – initially implemented by [22] and followed by [21,26,36,8,23] and many more by reproducing their code, this practice has become common for baselines in LLM Unlearning.

Another common sampling practice is **Cyclic Sampling** [25,12], in which all the retain samples are utilized by cycling forget samples. As in figure 2, a cyclic setting might have a retain sample unrelated to the forget sample. A drawback of this approach is the loss calculation of forget sample with unrelated retain sample. In this study, we introduce **Modular Entity-Level Unlearning (MELU)** strategy, in which during the unlearning process, each forget target is paired only with its respective retain samples.

3 Related Work

Current unlearning datasets include either direct (N_d) or indirect neighbor(N_{ind}) sets, but never both. **TOFU** [22] uses 200 synthetic authors as indirect neighbors (N_{ind}), plus 100 real authors and 117 facts, but omits interconnectivity [26] and ignores direct neighbors(N_d). **RWKU** [14] and **WPU** [21] include only a Direct Neighbor set plus a general knowledge set. **RWKU**’s focus on 200 high profile figures makes unlearning impractical because their large online footprints means pre-trained LLMs almost certainly have absorbed vast amounts of their data, making it difficult to unlearn; realistic unlearning requests involve individuals with moderate online presence. Datasets from [4,5] attempt to include both (N_d) and (N_{ind}) but they rely on bi-directional relationships for (N_d), requiring mutual links in their respective Wiki pages, creating a blind spot: e.g., "*Varchi was born in Florence*", Varchi’s page links to Florence, but Florence’s page does

³ <https://llmunlearningsemeval2025.github.io/>

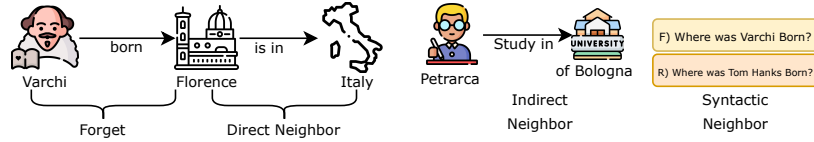


Fig. 1: Types of Neighbor sets and their connections to the forget sample

not link back to Varchi. Although Florence is directly connected and would influence the unlearning process, the bi-directional approach would exclude this from the retain set. Even though these benchmarks exist, LLM unlearning still lacks a standard protocol or methodology for building forget and retain sets, even as unlearning requests become common across applications. [30] provides a new direction on backdrops of the unlearning benchmark datasets. Post-Unlearning, they combine forget and retain queries and ask the model, only to find the model either recognizes them both as forget samples (*outputs IDK*) or retain samples (*outputs correct answers*).

In the *1:1 Sampling* **Method-a** limits the retain set during the data construction, and **Method-b** limits the retain set during unlearning, especially experiments conducted on [22] benchmark use various splits, fail to leverage the full retain set. For example, TOFU benchmark (4000 samples) has three splits of various forget set sizes (40, 200, 400), leading various retain set sizes (3.96k, 3.8k, 3.6k). TOFU authors unlearn for 5 epochs on these splits. For the largest split of 400 forget samples, at the maximum can attend only 2000 retain samples (if retain samples are sequentially chosen). Albeit, SemEVAL dataset uses **Method-a**, the winner of the competition [25] doesn't follow this approach, instead they follow 1:n-forget:retain approach (in a cyclic sequential unlearning process), making for each batch only for 1 forget sample and n retain samples are present.

In this paper, we extend the WPU [21] neighbor by adding indirect neighbors with syntactically similar Q&A's, and a dedicated test set. We further look into the sampling strategies provided by [22], 1:1 sampling, cyclic sampling in a batch unlearning scenario. We also introduce **Modular Entity-Level Unlearning (MELU)** strategy, in which each forget target is paired only with its respective retain samples. Our work neither aims to present a benchmark dataset nor an unlearning method, rather improve current best practices in creating LLM Unlearning Benchmark datasets and sampling.

4 Experimental Setup

4.1 Dataset Construction

In this work, we chose the Wikipedia Person Unlearning (WPU) [21] dataset, which consists of 100 forget targets, Direct neighbor set and general retain set.

The dataset is divided into forget_2, forget_20 and forget_100 parts, splitting the # of target entities to 2, 20 and 100. We chose forget_20 for the extension of Neighbor sets to (N_{ind}) , (N_s) and addition of a test set. The choice of WPU dataset comes from their approach in dataset construction. It was constructed by selecting the least popular Wikipedia people based on their views, these are the forget entities. They construct the D_r through incorporating (N_d) by scraping the hyperlinks that are connected to the Wikipedia page of the person (N_d) and General knowledge set by scraping the Wiki pages of popular people on Wikipedia (General Knowledge set). Other datasets use well known figures information, which might be difficult to unlearn, or a bi-directional approach for (N_d) , or synthetically created datasets. WPU stands as an ideal choice for our experiments, as the entities are not well known and has limited online presence providing a realistic unlearning situation.

Indirect Neighbor set (N_{ind}) creation - For Indirect connection we follow [22], and chose to find entities of similar profession. Finding similar profession entities for lesser known people was challenging as it requires scraping Google search suggestions, which were often unavailable or inconsistent. To overcome this, we used an LLM to generate similar profession names. Specifically, we chose LLaMA 3.3 70B model [10] for this task, since our unlearning experiments were conducted on the LLaMA 3.1 8B Instruct model [10]. We assumed that the models of the same family would likely share similar pre-training knowledge. We prompted (Appendix:A.1) the model to generate six names for each forget target, ending up with 120 indirect connections.

Once we had the Indirect connection entities, we scraped their Wikipedia data and used LLaMA 3.3 70B model to generate the Q&A's from it. We aimed for at least two questions per section, and instructed the LLM to follow an Interrogative syntactic structure. In total, we ended up with 1409 Q&A pairs. Then, for each forget target, we randomly picked five indirect connection entities to build the (N_{ind}) - giving us a total of 1144 Q&A pairs (Appendix:4).

Test set (D_t) creation - We construct a test with mix of multiple neighbors for evaluation. We use the remaining samples of (N_{ind}) for indirect connections and 200 random samples from the general knowledge set for the test set. To create the samples for (N_d) , we prompted LLaMA 3.3 70B to provide three new basic Q&A for every answer from forget set. If we are forgetting the link "Adrienne Monnier -> Paris", we made three Q&A's about Paris (Appendix:5). Finally, we created a test set with 738 Q&A pairs.

Due to the pre-structure of the WPU dataset [21], which includes (N_d) and general knowledge set, we were unable to create a standalone (N_s) dataset. Instead we generated the neighbor sets in a similar syntactic manner. To do this, in the Q&A generation prompt A.1, we provided the model to follow interrogative syntactic structure. We verified the syntactic similarity between the forget and retain Q&A pairs with edit distance algorithm [39], achieved a mean of 40% similarity.

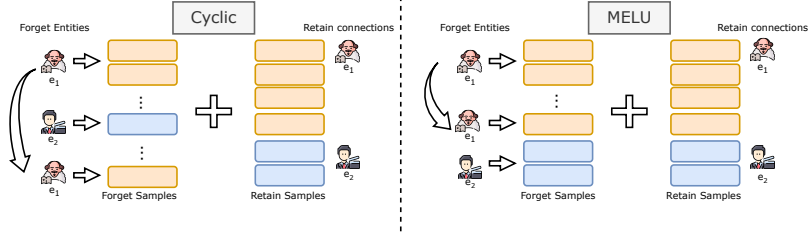


Fig. 2: MELU setting. In cyclic, each entities ($e_1, e_2..e_n$) forget samples are cycled on to unrelated ($e_3 \rightarrow e_1$) connections, In MELU entities ($e_1, e_2..e_n$) are cycled only onto their respective target connections ($e_1 \rightarrow e_1$).

4.2 Unlearning Methods

Before unlearning, we **fine-tuned** LLaMA 3.1 8B Instruct [10] model on all the datasets. We experiment with three unlearning algorithms- Gradient Difference [19] and Negative Preference Optimization [38] for Un-targeted unlearning and Direct Preference Optimization [27] for Targeted Unlearning. An Un-targeted unlearning does not contain a replacement sample for the forgetting sample, whereas targeted unlearning does and typically has phrases such as "I don't know".

We compare seven different settings. The initial three are on common data practices, and next four are common implementation practices. We have pool of $|D_f| = 98$, and $|D_r| = 1801$ samples. Given that all these experiments revolve around manipulating retain set, we select retain subsets D_r according to:

Data Practices⁴

1. **Direct-Neighbor** (N_d). Restrict D_r to only samples with the (N_d).
2. **Indirect-Neighbor** (N_{ind}). Restrict D_r to only samples with the (N_{ind}).
3. **Balanced**. We combine (N_d) and (N_{ind}) for our D_r . Since our (N_{ind}) is over sampled, we balance the dataset by selecting an equal number of samples as the (N_d) samples for each entity.

Sampling Practices

1. **1:1 seq**. i.e., We draw $|D_r| = |D_f|$ samples by selecting the top 98 D_r items, matching one retain for each forget. This is the method **a)** from section 2.3.
2. **1:1 random**. i.e., We draw $|D_r| = |D_f|$ samples by randomly selecting 98 D_r items for each epoch, matching one retain for each forget. This is the method **b)** from section 2.3.
3. **Cyclic**. Rotate through the full D_r pool sequentially until we collect 1801 samples - i.e. cycle the 98 sized $|D_f|$ across all the 1801 samples.

⁴ Given 1:1 sampling (both Sequence and Random) will not utilize the complete D_r during Unlearning, all the Data Practices experiments are conducted in Cyclic Implementation.

4. **Modular Entity-Level Unlearning (MELU)**. For each entity or forget target appearing the D_f , we include only retain samples that share that same entity. To do this, we cycle the forget samples of a forget target only over the retain samples of the same target (figure 2). We will be left with general knowledge set to which we randomly assign a sample from the D_f .

We adopt LoRA[11] for all our experiments. For finetuning, rank = 64, α = 128, batch size = 32 for 10 epochs. For unlearning, rank = 8, α = 16, batch size = 8⁵ for 4 epochs. All the experiments were conducted on 2 x 40GB A100 GPUs. Full algorithmic details are in Appendix:A.3.

4.3 Assessment

Unlearning behavior is best assessed with the use of multiple metrics [22]. We employ three distinct metrics and aggregate them to compute two scores: **Forget Efficacy** and **Model Utility**. In line with prior works [22,36,38], we employ **ROUGE-L** (verbatim memorization with word-level match), **Conditional Probability** (ground truth likelihood) and **Cosine Similarity** (Semantic Similarity). To calculate *Forget Efficacy*, we calculate $1 - \text{Arithmetic mean}$ of these metric on D_f and for *Model Utility* we calculate a harmonic mean of these metrics on D_r .

5 Results and Discussion

Baselines We computed Forget Efficacy (on forget set) and Model Utility (on test set) on the base model before unlearning. We use these results as baselines. After applying the unlearning algorithms, for a fair comparison, we use Forget Efficacy and Model Utility on Test set (MU-T), given that the $|D_r|$ changes based on the setting. The test set is balanced and will be used to understand the Model Utility. The base model has a low Forget Efficacy (FE = 0.30) and high model utility (ME-T = 0.73). A good unlearned model should have a higher FE (up to 1) and MU-T closer to the baseline ± 5 . We also compute MMLU [34] scores to understand the general Model utility, per target FE and MU-T for granular understanding and token diversity with **Distinct-N** [16] to understand the diversity of the generated outputs.

5.1 Evaluation of Unlearning Data Practices

Direct vs Indirect: Interestingly, both GD and NPO show (figure 3 and table 1) a drop in FE when moving from Direct (N_d) to Indirect (N_{ind}) neighbor sets, contrary to our expectations. Since, (N_d) is smaller in size (# 364 samples + # 293 general knowledge) compared to (N_{ind}) (#1144 + #293), we anticipated forget set would be revised more frequently during unlearning. However, (N_d)

⁵ Batch size of 8 is maintained for every experiment by the aggregation of gradients over 8 samples, even when the hardware limitations prohibit batch size of 8.

neighbor with such a small neighbor set outperforms (N_{ind}) in terms of FE. In contrast, DPO follows the expected trend, showing higher FE with the larger (N_{ind}). However, **MU-T is consistently higher** for (N_{ind}), indicates, larger and more diverse sets preserve general model performance. On the other hand, Balanced fails to achieve better FE and MU-T.

From the **Token Diversity** 9a on the forget and test sets, we find that (N_d) is lower than the (N_{ind}) neighbors on the test set across all the unlearning algorithms. For GD, we see an exponential drop in forget set diversity in indirect and balanced. For DPO and NPO, we see an increment in token diversity from indirect to balanced. But both GD and DPO fail to maintain token diversity (even in implementation settings), this is because of GD’s ‘*Degeneration Behavior*’ and DPOs ‘*I don’t know*’ phrases. NPO exhibits more favorable behavior with high token diversity. This is likely due to its bounded objective, preventing model collapse.

Per target FE and MU-T show that GD, performs really well at forgetting with direct connections, but fails significantly at MU-T (10 targets are below 0.20 for MU-T). We find a similar situation with balanced, where the FE is higher and MU-T is lower. Although, indirect doesn’t achieve same level of forgetting as direct, it always maintains > 0.85 FE on all targets and maintains MU-T up to 0.65 (0.08 shy from baseline). With preference based methods, we find a gradual increment in MU-T from direct to indirect to balanced. A strange phenomenon was observed that some targets such as “*Ted Kooser*” was harder to forget for both DPO and NPO (except direct case for NPO). With memorization scores (Appendix:Fig11), we find that these targets are highly memorized than others.

MMLU scores on GD are inconsistent. Although they are not exponential drops or highs, but we find an increase in MMLU accuracy (Appendix:Fig8) on direct and balanced dataset experiments ($\approx +1.3\%$). This is an unusual behavior. Where as, preference based methods show a stable accuracy. Cyclic and MELU, provide stable MMLU scores across all the models, showing with proper implementation unlearning can be stable.

5.2 Evaluation of Unlearning Sampling Practices

Standard **1:1 sampling** (sequential and random) fails to produce meaningful forgetting (low FE) yet preserve MU-T across all the unlearning algorithms. Although, increase in number of epochs might improve the forgetting⁶, we already achieve better stability (FE and MU-T) with cyclic and MELU with the same number of epochs.

⁶ To test this we conducted a run of DPO with 1:1 random sampling by continuously increasing the epochs. At epoch 100 we achieved 0.79 FE and 0.78 MU-T.

Stability with MELU Both Cyclic and MELU perform significantly better than standard 1:1 sampling. They maintain stable performance across all the unlearning algorithms and maintain accuracy on MMLU and token diversity. MELU, in particular, outperforms cyclic under DPO, boosting FE by 12% while maintaining MU-T. In NPO, MELU provides a small improvement from cyclic. But at the **per-target** performance, MELU holds a better FE and MU-T on all the targets for preference based methods. Under DPO, MELU increases the number of targets with $FE > 0.9$ (from 1 in cyclic to 3), while maintaining high $MU-T \geq 0.8$ for the majority. In case of Amy Clampitt, FE improves by $\approx 20\%$. For GD, MELU setup achieves stable FE (≈ 0.9) across most entities and provides higher MU-T across targets. Even in NPO, where overall FE grows slowly, MELU maintains MU-T while achieving reasonable FE. In cases of harder targets such as "*Ann Brashares*" and *Ted Kooser*, both cyclic and MELU perform well and forget (> 0.50) better than 1:1 sampling (< 0.10). This improved stability of MELU can be attributed to a more consistent learning signal. In cyclic sampling, the model is subjected to high-variance gradients due to unrelated forget-retain pairs. MELU, by maintaining a relevancy between forget retain with lower variance per batch, could be leading to a stable performance.

Overall MELU provides

1. **High and Stable FE:** approaching or exceeding 0.85 for DPO and GD, and maintaining competitive scores in NPO.
2. **Minimal degradation in MU-T:** consistently close to the baseline (0.73), even slightly exceeding it for some algorithms (e.g., NPO).

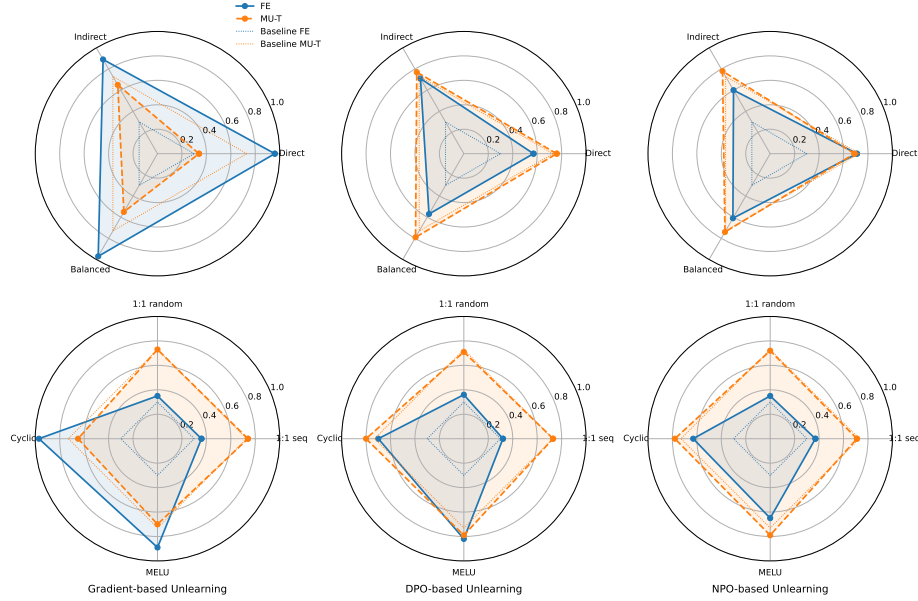


Fig. 3: FE and MU-T comparison results. Top row provides the data practices results and bottom row provides implementation practices results.

6 Conclusion

Our study demonstrates that the composition of the retain set is a critical, yet often under-looked factor in LLM Unlearning. Our findings show that relying solely on any single neighbor set is suboptimal and not the best practice. By including a diverse range of neighbors - we can improve the balance between forget efficacy and model utility. Furthermore, we show that the standard 1:1 sampling is an inefficient approach and when implementing unlearning, cyclic and Modular Entity-Level Unlearning (MELU) provides more stability. Albeit, we couldn't answer why these setups provides more stability, a research that could be looked into is if this caused due to the repetition of forget samples in the unlearning process (It is well-known that during the pretraining, memorization of a sample is correlated with its repetition in the corpus [2]). Our work also proves some of the already proposed problems in the unlearning literature, such as Gradient based approach's instability [33,35], and how some targets are harder to forget because of the frequency of their knowledge and memorization (Appendix:11) in pre-trained or downstream task data [15]. We hope our work inspires researchers to look into methods to construct more diverse and realistic unlearning benchmarks and unlearning algorithm implementation techniques.

For future works, We aim to conduct a rigorous comparative analysis against other model families, more unlearning algorithms and conduct a deeper evaluation, especially with consideration of sample memorization. Additionally, we aim

to do a comparative analysis with Sequential Unlearning and also incorporating MELU into sequential unlearning.

7 Limitations

A key limitation of our study is the indirect neighbors and test sets generation with sister model. This limits the generalizability of our findings and requires expanding our experiments to other model families. Our MELU setup assumes that forget and retain samples are sufficiently distinct to be reliably separated. In real-world scenarios, however, such clear boundaries may not always exist, especially when entities share overlapping attributes or contexts, albeit one can enforce Knowledge graphs to define these clear connections[26]. Additionally, we do not deep dive into instability and stability issues such as GD and NPO’s better forgetting on direct neighbors and not with indirect neighbors. In contrast DPO acts opposite, this can be further looked into especially through the lens of explainability approaches on pre and post unlearning. Same with MELU and cyclic settings stable performances, a rigorous work needs to be done towards addressing it. Another limitation is extension of general utility with HellaSwag [37], ARC [6] etc. Our experiments cover only a few unlearning algorithms with batch unlearning. While we propose MELU, we lack a direct comparative analysis to sequential unlearning [12] setups. Finally, because WPU [21] already includes direct and general neighbors, we could not construct a full syntactic neighbor set, leaving it unexplored.

Acknowledgments

This work has been partially supported by the EU EIC project EMERGE (Grant No. 101070918).

References

1. Brown, T.B., Mann, B., Ryder, e.a.: Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS ’20, Curran Associates Inc., Red Hook, NY, USA (2020)
2. Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., Zhang, C.: Quantifying memorization across neural language models. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=TatRHT_1cK
3. Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, Ú., Oprea, A., Raffel, C.: Extracting training data from large language models. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 2633–2650. USENIX Association (Aug 2021), <https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting>

4. Chang, H., Lee, H.: Which retain set matters for LLM unlearning? a case study on entity unlearning. In: Che, W., Nabende, J., Shutova, E., Pilehvar, M.T. (eds.) Findings of the Association for Computational Linguistics: ACL 2025. pp. 5966–5982. Association for Computational Linguistics, Vienna, Austria (Jul 2025). <https://doi.org/10.18653/v1/2025.findings-acl.310>, <https://aclanthology.org/2025.findings-acl.310/>
5. Choi, M., Rim, D., Lee, D., Choo, J.: Opt-out: Investigating entity-level unlearning for large language models via optimal transport. In: Che, W., Nabende, J., Shutova, E., Pilehvar, M.T. (eds.) Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 28280–28297. Association for Computational Linguistics, Vienna, Austria (Jul 2025). <https://doi.org/10.18653/v1/2025.acl-long.1371>, <https://aclanthology.org/2025.acl-long.1371/>
6. Chollet, F., Knoop, M., Kamradt, G., Landers, B.: Arc prize 2024: Technical report (2025), <https://arxiv.org/abs/2412.04604>
7. Dorna, V., Mekala, A., Zhao, W., McCallum, A., Kolter, J.Z., Maini, P.: OpenUnlearning: A unified framework for llm unlearning benchmarks. <https://github.com/locuslab/open-unlearning> (2025), accessed: February 27, 2025
8. Fan, C., Liu, J., Lin, L., Jia, J., Zhang, R., Mei, S., Liu, S.: Simplicity prevails: Rethinking negative preference optimization for LLM unlearning (2025), <https://openreview.net/forum?id=Pd3jVGTacT>
9. Golatkar, A., Achille, A., Soatto, S.: Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks . In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9301–9309. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2020). <https://doi.org/10.1109/CVPR42600.2020.00932>, <https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.00932>
10. Grattafiori, A., Dubey, A., et al, A.J.: The llama 3 herd of models (2024), <https://arxiv.org/abs/2407.21783>
11. Hu, E.J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022), <https://openreview.net/forum?id=nZeVKeeFYf9>
12. Jang, J., Yoon, D., Yang, S., Cha, S., Lee, M., Logeswaran, L., Seo, M.: Knowledge unlearning for mitigating privacy risks in language models. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 14389–14408. Association for Computational Linguistics, Toronto, Canada (Jul 2023). <https://doi.org/10.18653/v1/2023.acl-long.805>, <https://aclanthology.org/2023.acl-long.805/>
13. Ji, J., Liu, Y., Zhang, Y., Liu, G., Kompella, R.R., Liu, S., Chang, S.: Reversing the forget-retain objectives: An efficient LLM unlearning framework from logit difference. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024), <https://openreview.net/forum?id=tYdR1lTWqh>
14. Jin, Z., Cao, P., Wang, C., He, Z., Yuan, H., Li, J., Chen, Y., Liu, K., Zhao, J.: RWKU: Benchmarking real-world knowledge unlearning for large language models. In: The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2024), <https://openreview.net/forum?id=wOmtZ5FgMH>

15. Krishnan, A., Reddy, S., Mosbach, M.: Not all data are unlearned equally. In: Second Conference on Language Modeling (2025), <https://openreview.net/forum?id=Kd971fFfTu>
16. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 110–119. Association for Computational Linguistics, San Diego, California (Jun 2016). <https://doi.org/10.18653/v1/N16-1014>, <https://aclanthology.org/N16-1014/>
17. Li, N., Pan, A., Gopal, A., Yue, S., Berrios, e.a.: The WMDP benchmark: Measuring and reducing malicious use with unlearning. In: Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 28525–28550. PMLR (21–27 Jul 2024), <https://proceedings.mlr.press/v235/li24bc.html>
18. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), <https://aclanthology.org/W04-1013/>
19. Liu, B., Liu, Q., Stone, P.: Continual learning and private unlearning (2022), <https://arxiv.org/abs/2203.12817>
20. Liu, S., Yao, Y., Jia, J., et al.: Rethinking machine unlearning for large language models. *Nature Machine Intelligence* **7**, 181–194 (2025). <https://doi.org/10.1038/s42256-025-00985-0>, <https://doi.org/10.1038/s42256-025-00985-0>
21. Liu, Y., Zhang, Y., Jaakkola, T., Chang, S.: Revisiting who’s harry potter: Towards targeted unlearning from a causal intervention perspective. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. pp. 8708–8731. Association for Computational Linguistics, Miami, Florida, USA (Nov 2024). <https://doi.org/10.18653/v1/2024.emnlp-main.495>, <https://aclanthology.org/2024.emnlp-main.495/>
22. Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z.C., Kolter, J.Z.: TOFU: A task of fictitious unlearning for LLMs. In: First Conference on Language Modeling (2024), <https://openreview.net/forum?id=B41hNBwLo>
23. Mekala, A., Dorna, V., Dubey, S., Lalwani, A., Koleczek, D., Rungta, M., Hasan, S., Lobo, E.: Alternate preference optimization for unlearning factual knowledge in large language models. In: Proceedings of the 31st International Conference on Computational Linguistics. pp. 3732–3752. Association for Computational Linguistics, Abu Dhabi, UAE (Jan 2025), <https://aclanthology.org/2025.coling-main.252/>
24. Miranda, M., Ruzzetti, E.S., Santilli, A., Zanzotto, F.M., Bratières, S., Rodolà, E.: Preserving privacy in large language models: A survey on current threats and solutions. *Transactions on Machine Learning Research* (2025), <https://openreview.net/forum?id=Ss9MTTN70L>
25. Premptis, I., Lymperaïou, M., Filandrianos, G., Menis Mastromichalakis, O., Voulodimos, A., Stamou, G.: AILS-NTUA at SemEval-2025 task 4: Parameter-efficient unlearning for large language models using data chunking. In: Rosenthal, S., Rosá, A., Ghosh, D., Zampieri, M. (eds.) Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025). pp. 1383–1405. Association for Computational Linguistics, Vienna, Austria (Jul 2025), <https://aclanthology.org/2025.semeval-1.184/>
26. Qiu, X., Shen, W.F., Chen, Y., Kurmanji, M., Cancedda, N., Stenetorp, P., Lane, N.D.: How data inter-connectivity shapes llms unlearning: A structural unlearning perspective (2025), <https://arxiv.org/abs/2406.16810>

27. Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C.D., Finn, C.: Direct preference optimization: your language model is secretly a reward model. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23, Curran Associates Inc., Red Hook, NY, USA (2023)
28. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992. Association for Computational Linguistics, Hong Kong, China (Nov 2019). <https://doi.org/10.18653/v1/D19-1410>, <https://aclanthology.org/D19-1410/>
29. Staab, R., Vero, M., Balunovic, M., Vechev, M.: Beyond memorization: Violating privacy via inference with large language models. In: The Twelfth International Conference on Learning Representations (2024), <https://openreview.net/forum?id=kmn0BhQk7p>
30. Thaker, P., Hu, S., Kale, N., Maurya, Y., Wu, Z.S., Smith, V.: Position: Llm unlearning benchmarks are weak measures of progress (2025), <https://arxiv.org/abs/2410.02879>
31. Tirumala, K., Markosyan, A.H., Zettlemoyer, L., Aghajanyan, A.: Memorization without overfitting: Analyzing the training dynamics of large language models. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022), <https://openreview.net/forum?id=u3vEuRr08MT>
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 6000–6010. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
33. Wang, Q., Zhou, J.P., Zhou, Z., Shin, S., Han, B., Weinberger, K.Q.: Rethinking LLM unlearning objectives: A gradient perspective and go beyond. In: The Thirteenth International Conference on Learning Representations (2025), <https://openreview.net/forum?id=huo8MqVH6t>
34. Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., Li, T., Ku, M., Wang, K., Zhuang, A., Fan, R., Yue, X., Chen, W.: Mmlu-pro: A more robust and challenging multi-task language understanding benchmark (2024), <https://arxiv.org/abs/2406.01574>
35. Yao, Y., Xu, X., Liu, Y.: Large language model unlearning. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024), <https://openreview.net/forum?id=8Dy42ThoNe>
36. Yuan, X., Pang, T., Du, C., Chen, K., Zhang, W., Lin, M.: A closer look at machine unlearning for large language models. In: The Thirteenth International Conference on Learning Representations (2025), <https://openreview.net/forum?id=Q1MHvGmhyT>
37. Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., Choi, Y.: HellaSwag: Can a machine really finish your sentence? In: Korhonen, A., Traum, D., Màrquez, L. (eds.) Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 4791–4800. Association for Computational Linguistics, Florence, Italy (Jul 2019). <https://doi.org/10.18653/v1/P19-1472>, <https://aclanthology.org/P19-1472/>
38. Zhang, R., Lin, L., Bai, Y., Mei, S.: Negative preference optimization: From catastrophic collapse to effective unlearning. In: First Conference on Language Modeling (2024), <https://openreview.net/forum?id=MXLBXjQkmb>

39. Zhang, S., Hu, Y., Bian, G.: Research on string similarity algorithm based on levenshtein distance. In: 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). pp. 2247–2251 (2017). <https://doi.org/10.1109/IAEAC.2017.8054419>

A Appendix

A.1 Prompts

Q&A Generation *This prompt was used to extract Q&As from the wiki pages.*

LLaMA 3.3-70B

```
#system_prompt\\
You are an expert teacher, who can create questions and answers
from a given context.
Given the user wikipedia page context about {#
domain_person_name},

please provide as many questions and answers possible from it.

For each section, provide at least 2 questions and answers.

The question and answers should follow the Interrogative
syntactic structure,

The questions should be on their birth, family background,
education, career, achievements and other relevant topics.

The output should be in JSON format with the following keys:

\\{
  "name": name of the person,
  "question1": question1,
  "answer1": answer1,
  "section" : part of the wikipedia section,
  "difficulty" : difficulty of the question,
  "question2": question2,
  \dots
\\}

Please be precise with the question and answer. Do not generate
any other text.

#prompt\\
{# content}
```

Indirect Connection generation *This prompt was used to generate six Indirect connections for a target.*

```

LLaMA 3.3-70B

#prompt\\
For each name in the list, provide me 6 names that belong to the
same domain as them (for example, if they are authors please
provide authors similar as them). The output should be in a
dictionary to make it into a dataframe.

['Benedetto Varchi', 'Wilhelm Wattenbach', 'Elsa Triolet',
 'Theopompus', 'Heinrich Ritter', 'Adrienne Monnier', 'Ann
 Brashares', 'Hartmann von Aue', 'Jorge Semprún', 'Giovanni
 Battista Casti', 'Najaf Daryabandari', 'Heinz Erhardt',
 'Rudolf Christoph Eucken', 'Paul Gerhardt', 'Moshe
 Greenberg', 'Amy Clampitt', 'Ted Kooser', 'Alfred Vogel',
 'Siegfried Lenz', 'Philip Stanhope, 5th Earl Stanhope']

```

A.2 Dataset

A detailed pipeline in creating the indirect connections and its relevant test set samples are provided in the Figure 4. First, we use the prompt A.1 to generate 6 entities for each target. Followed by we scrape their wiki pages and generate Q&As with the LLaMA 3.3 70B model A.1 in an Interrogative syntactic structure manner to maintain the (N_s) neighbor dataset.

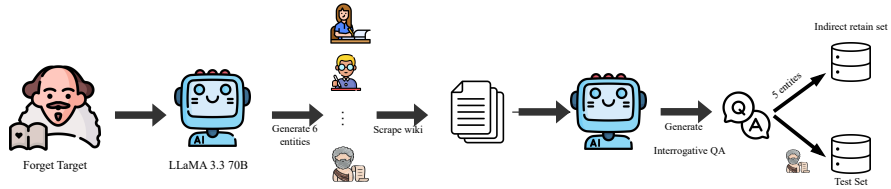


Fig. 4: Pipeline in creating Indirect connections for retain and test set

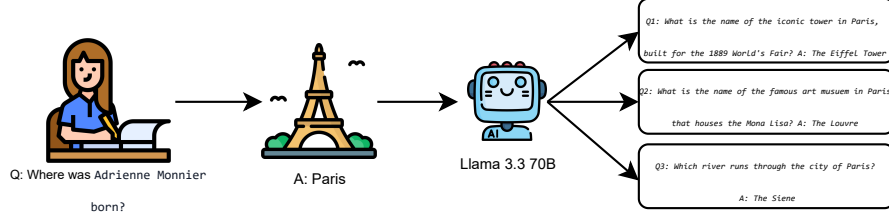


Fig. 5: Generation of Test set samples for the Direct Neighbors

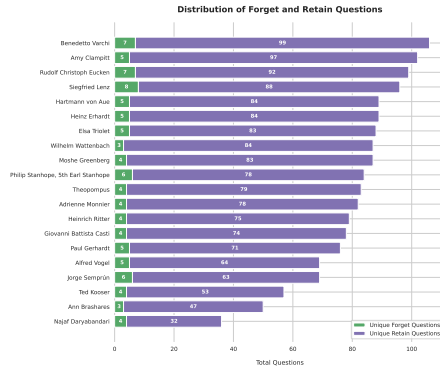


Fig. 6: Composition of Forget-Retain samples per target

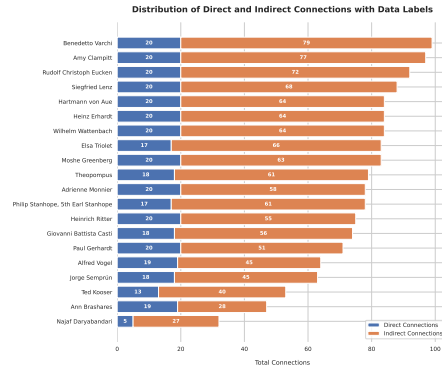


Fig. 7: Composition of Direct and Indirect retain samples per target

A.3 Experimental setup

Model Finetuning: Before the application of unlearning, we initially finetune the model on the all the datasets we have i.e., D_f, D_r, D_t . We use the questions as prompts and conduct a Supervised Fine-tuning on the datasets. Given $D_{ft} = D_f + D_r + D_t$, and its samples (x, y) , x is question and y is the answer. A pair $p_i = p(x_i, y_i) \in D_{ft}$ and $y_1, ..y_T$ are the answer tokens, we calculate Negative-Log-Likelihood (NLL) loss for p_i

$$\mathcal{L}(y | x; \theta) = \text{NLL}(y | x; \theta) = - \sum_{t=1}^T \log p(y_t | x, y_{<t}; \theta) \quad (1)$$

Unlearning Methods:

Gradient Ascent (GA) is the most straightforward unlearning technique proposed for the Un-targeted unlearning. It's main idea is to maximize the loss as opposed to the training objective of minimization by negating the loss. In our work, we do not implement this. Due to its nature of negation, the maximization

becomes unbounded leading to catastrophic collapse. The predicted loss $l(y|x; \theta)$ on forget set can be written as:

$$\mathcal{L}_{GA}(D_f; \theta) = -\mathcal{L}(y_f | x_f; \theta) \quad (2)$$

Gradient Difference (GD) proposed by [19] to mitigate the issues of Gradient ascent. It builds on the concept of Gradient Ascent, but not only aims to maximize the loss on forget set D_f , simultaneously minimizes the loss on the retain set D_r . This maintains the balance of forgetting and retaining. The loss function can be written as:

$$\mathcal{L}_{GD}(\theta) = -\mathcal{L}(D_f; \theta) + \mathcal{L}(D_r; \theta) \quad (3)$$

Direct Preference Optimization (DPO) is proposed by [27] and was first used by [22], treats unlearning as a preference optimization problem by applying the standard DPO loss. This technique uses *Targeted Unlearning*, making a necessity of replacement responses like "I don't know". Alike the standard DPO approach, we use "I don't know" responses as positive samples and forget set as negative samples to guide the model's response. For the implementation of DPO, we convert the forget set to a preference dataset containing winning responses and refusal responses. Preference dataset $D_p = (x_i, y_{i,win}, y_{i,lose}), i \in |D_f|$, where $y_{i,win}$ are randomly chosen from a subset of "I don't know" phrases, and $y_{i,lose}$ are the forget targets. The DPO loss can be calculated as:

$$\mathcal{L}_{DPO,\beta}(\theta) = -\mathbb{E}_{D_p} \left[\log \sigma \left(\beta \log \frac{p(y_{win} | x; \theta)}{p(y_{win} | x; \theta_{ref})} - \beta \log \frac{p(y_{lose} | x; \theta)}{p(y_{lose} | x; \theta_{ref})} \right) \right] \quad (4)$$

where σ is the sigmoid function and β is the inverse temperature controlling the preference strength. We use $\beta = 0.1$ for all our experiments. Provided, we have a retain set, we utilize the code implementation provided by [7], can be calculated as follows:

$$\mathcal{L}_{DPO+retain} = \alpha \mathcal{L}_{DPO,\beta}(\theta) + \gamma \mathcal{L}(D_r; \theta) \quad (5)$$

Where α and γ are hyperparameter to control the strength of DPO loss and NLL. For our experiments, both α and γ is always 1.

Negative Preference Optimization (NPO) - proposed by [38], is an inspiration of DPO, a variant that uses only the negative responses from the D_f , disregarding the y_{win} making it an Un-targeted Unlearning. For the implementation, we ignore the "I don't know" responses and provide only (x_f, y_f) . Followed by, we calculate retain loss similarly as in DPO + retain with the same hyperparameters values (β, α, γ) .

$$\mathcal{L}_{NPO,\beta}(\theta) = -\frac{2}{\beta} \mathbb{E}_{D_p} \left[\log \sigma \left(-\beta \log \frac{p(y_{lose} | x; \theta)}{p(y_{lose} | x; \theta_{ref})} \right) \right] \quad (6)$$

$$\mathcal{L}_{\text{NPO+retain}} = \alpha \mathcal{L}_{\text{NPO},\beta}(\theta) + \gamma \mathcal{L}(D_r; \theta) \quad (7)$$

Evaluation Metrics: ROUGE (R) quantifies the word-level overlap between the model’s output and the ground-truth answer. We compute the ROUGE-L [18] score between the generated response $g(x; \theta_*)$ and the ground-truth answer y , written as $\text{ROUGE} - L(g(x; \theta_*), y)$. ROUGE-L provides the longest sequence overlap and the verbatim memory of the Unlearned Model $(M; \theta_*)$.

Cosine Similarity (CS) measures the semantic similarity of the model’s output against the ground-truth. We follow [36] setup, embed both with Sentence-BERT [28], calculate the cosine similarity and truncate the values less than 0.

$$\max(\cos(g(x; \theta_*), y), 0)$$

Probability (P) defines the average likelihood assigned to each token given a question and its ground truth answer i.e., (x, y) . Following [22], we compute normalized conditional probability as

$$\mathcal{P}(y \mid x) = \frac{1}{T} \sum_{t=1}^T p(y_t \mid x \circ y_{<t}; \theta_*)$$

A.4 Results

Table 1: Experimental results.

Dataset Practices					
Method	FE \uparrow	MU-T \uparrow	PPL-F \downarrow	PPL-T \downarrow	MMLU %
Pre-Unlearning	0.30	0.73	38.76	37105	12.42
Gradient-based (Un-Targeted)					
GA	0.44	0.67	657294.87	242062.34	12.47
Direct	0.96	0.34	3.09×10^{82}	3.28×10^{80}	13.60
Indirect	0.89	0.65	1.27×10^{90}	1.79×10^{82}	8.40
Balanced	0.97	0.55	2.24×10^{85}	1.79×10^{82}	13.29
DPO-based (Targeted)					
DPO	0.70	0.47	16643	3098	12.21
Direct	0.57	0.76	1.82×10^4	158.72	12.27
Indirect	0.71	0.77	5.84×10^7	180.30	12.61
Balanced	0.57	0.79	1.71×10^5	142.58	12.35
NPO-based (Un-Targeted)					
NPO	0.30	0.73	38.76	37105	12.37
Direct	0.71	0.69	4.68×10^{22}	1.002×10^{17}	12.57
Indirect	0.60	0.78	3.17×10^{18}	126.47	12.37
Balanced	0.61	0.74	2.6×10^{18}	3153.19	12.63

Sampling Practices					
Method	FE \uparrow	MU-T \uparrow	PPL-F \downarrow	PPL-T \downarrow	MMLU %
Gradient-based (Un-Targeted)					
1:1 seq	0.36	0.74	3.7×10^4	6.11×10^4	12.46
1:1 random	0.35	0.73	22521	32751	12.31
Cyclic	0.97	0.65	1.80×10^{86}	3.08	13.26
MELU	0.89	0.70	3.18×10^{90}	15.98	13.42
DPO-based (Targeted)					
1:1 seq	0.32	0.73	124.55	3.0×10^3	12.33
1:1 random	0.36	0.71	65.90	2539.52	12.28
Cyclic	0.70	0.80	2.57×10^7	118.54	12.36
MELU	0.82	0.79	5.8×10^{14}	87.71	12.38
NPO-based (Un-Targeted)					
1:1 seq	0.37	0.71	1655.97	5.33×10^5	12.43
1:1 random	0.35	0.72	14545.83	78544	12.36
Cyclic	0.63	0.78	1.60×10^{18}	35.15	12.24
MELU	0.65	0.79	7.86×10^{21}	54.03	12.41

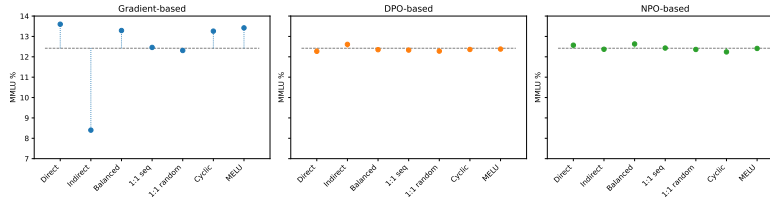
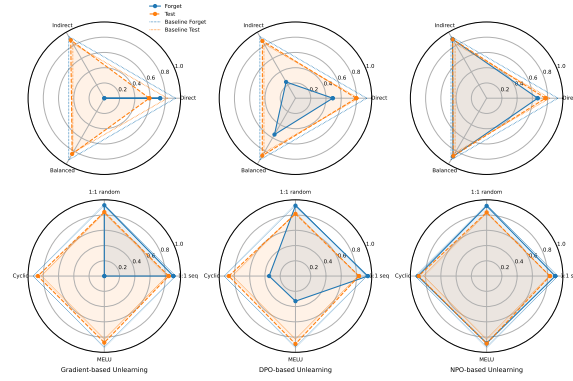
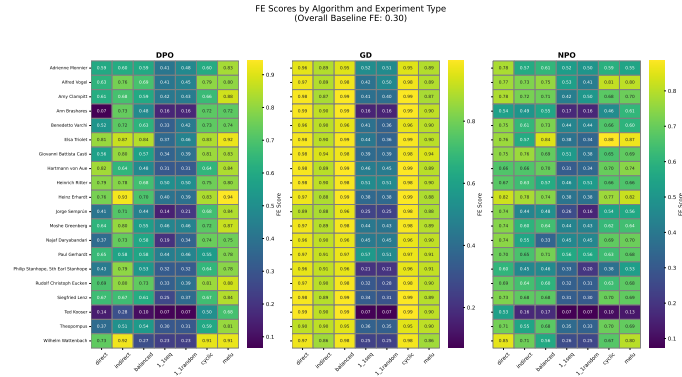


Fig. 8: General Model Utility (MMLU) across all experiments. Baseline accuracy is 12.42%.

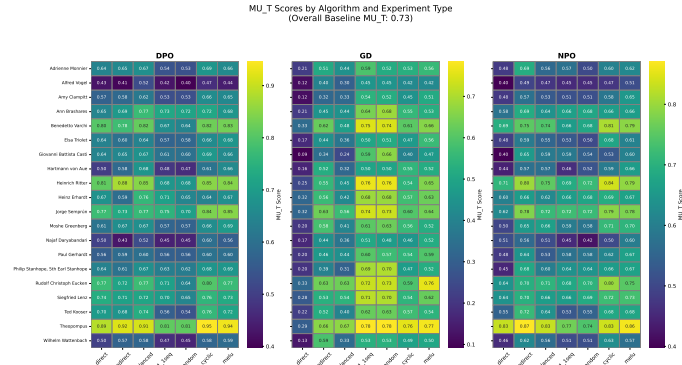


(a) Token diversity of the unlearned models. Top row: data practices; bottom row: implementation practices.

Heat maps of per-entity FE and MU-T



(b) Forget Efficacy (FE).



(c) Model Utility Test (MU-T).

Fig. 9: Token diversity and per-entity metrics.

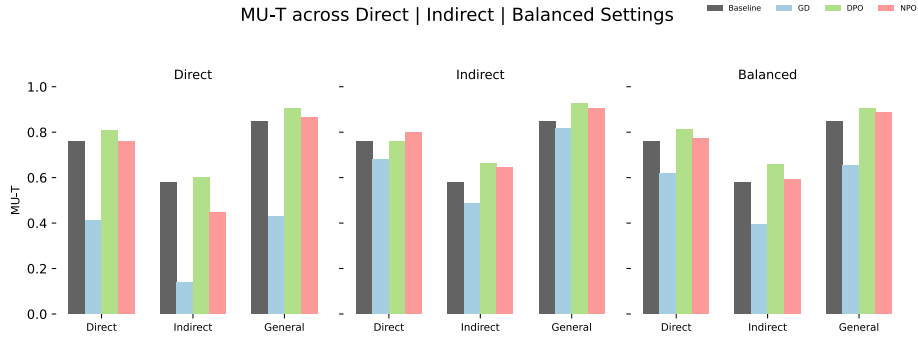


Fig. 10: MU-T across various data practices. We find Gradient Based method consistently performs bad in all the Data settings, where as NPO and DPO holds the MU-T well.

A.5 Memorization

We calculated each entities memorization with Exact Memorization (EM) score following [31], which is often used in Unlearning research to define/compute the success of forgetting in LLMs. Given we have multiple samples for each entity, we compute their Average EM score. We find that few samples such as *Ted Kooser*, *Philip Stanhope*, *Ann Brashes* etc are highly memorized and were harder to forget in our experiments.



Fig. 11: Memorization Scores of Each Entity