DP2Unlearning: An Efficient and Guaranteed Unlearning Framework for LLMs

Tamim Al Mahmud*, Najeeb Jebreel, Josep Domingo-Ferrer, David Sánchez

Universitat Rovira i Virgili,
Department of Computer Engineering and Mathematics,
CYBERCAT-Center for Cybersecurity Research of Catalonia,
Av. Països Catalans 26, 43007 Tarragona, Catalonia

Abstract

Large language models (LLMs) have recently revolutionized language processing tasks but have also brought ethical and legal issues. LLMs have a tendency to memorize potentially private or copyrighted information present in the training data, which might then be delivered to end users at inference time. When this happens, a naive solution is to retrain the model from scratch after excluding the undesired data. Although this *guarantees* that the target data have been forgotten, it is also prohibitively expensive for LLMs. Approximate unlearning offers a more efficient alternative, as it consists of *ex post* modifications of the trained model itself to prevent undesirable results, but it lacks forgetting guarantees because it relies solely on empirical evidence. In this work, we present *DP2Unlearning*, a novel LLM unlearning framework that offers formal forgetting guarantees at a significantly lower cost than retraining from scratch on the data to be retained. DP2Unlearning involves training LLMs on textual data protected using ϵ-differential privacy (DP), which later enables efficient unlearning with the guarantees against disclosure associated with the chosen ϵ. Our experiments demonstrate that DP2Unlearning achieves similar model performance post-unlearning, compared to an LLM retraining from scratch on retained data –the gold standard exact unlearning– but at approximately half the unlearn-

*Corresponding author

Email addresses: tamimal.mahmud@urv.cat (Tamim Al Mahmud), najeeb.jebreel@urv.cat (Najeeb Jebreel), josep.domingo@urv.cat (Josep Domingo-Ferrer), david.sanchez@urv.cat (David Sánchez)

Published in *Elsevier Neural Networks Journal*, 18 July 2025,
Access: https://doi.org/10.1016/j.neunet.2025.107879

ing cost. In addition, with a reasonable computational cost, it outperforms approximate unlearning methods at both preserving the utility of the model post-unlearning and effectively forgetting the targeted information.

The code of our experiments is available at
https://github.com/tamimalmahmud/LLM-Unlearning/tree/main/DP2Unlearning.

Keywords: LLM Unlearning, Exact Unlearning, Approximate Unlearning, Differential Privacy, Privacy-preserving LLM.

---

1. Introduction

Thanks to training on massive text corpora, large language models (LLMs) (Achiam et al., 2023; Gemini et al., 2023; Liu et al., 2024) have transformed the landscape of natural language processing (NLP), excelling in various tasks such as question answering (Khashabi et al., 2020), translation (Lewis et al., 2020), and text generation (Lewis et al., 2020), as well as more complex applications such as education (Malinka et al., 2023) and recommendation (Manzoor et al., 2024).

Despite their potential, LLMs pose ethical risks (Weidinger et al., 2022). Their ability to memorize data seen during training (Tirumala et al., 2022; Carlini et al., 2023) can lead to the unintentional generation of *private information* (Lukas et al., 2023; Carlini et al., 2023) or the reproduction of *copyrighted content* (Chang et al., 2023; Karamolegkou et al., 2023). For example, (Carlini et al., 2021) have extracted hundreds of verbatim text sequences from GPT-2 training examples, which contained personally identifiable information (PII) such as names, phone numbers, and email addresses. In (Li et al., 2023), it was shown that despite the measures taken to prevent the generation of sensitive content by OpenAI's ChatGPT and the ChatGPT enhanced Bing search engine, adversarially designed prompts could still allow PII extraction from these models. (Karamolegkou et al., 2023) found that LLMs memorize many copyrighted text fragments, including complete descriptions of LeetCode problems. Recently, proprietary algorithms have revealed large-scale verbatim reproduction of copyrighted material by LLMs, including content from NYT articles, works by Ta-Nehisi Coates and Stephen King, academic articles, song lyrics, and business publications (Hunt, 2024).

Legal frameworks such as the GDPR (Voigt and Von dem Bussche, 2017) in the EU and the CCPA (Department of Justice, 2024) in the US have been established to protect privacy and intellectual property in AI systems. The

GDPR emphasizes the Right to Be Forgotten (RTBF) for prompt deletion of personal data, while copyright laws balance creator rights with fair use in the US (U.S. Copyright Office, 2018) and quotation rights in the EU (The European Parliament, 2019). All of this presents a pressing challenge for LLM managers.

A naive approach to *forget* memorized private and copyright protected information from trained LLMs involves retraining the LLM from scratch after excluding the data to be forgotten. Although this approach provides forgetting guarantees, it is impractical for LLMs because the computational expense of processing each forgetting request is prohibitively high.

Machine unlearning (Jang et al., 2023; Yao et al., 2023; Maini et al., 2024) is emerging as a promising approach to achieve efficient forgetting. It refers to the process of selectively forgetting specific knowledge learned by an LLM without affecting unrelated knowledge. Based on their guarantee of forgetting, the unlearning methods can be categorized into *exact unlearning* and *approximate unlearning* (Xu et al., 2024). Exact unlearning methods ensure complete forgetting of unwanted data (Bourtoule et al., 2021; Hu et al., 2024), but are not practical for LLM due to their significant computational time and storage requirements. On the other hand, approximate unlearning provides a more efficient alternative, employing various heuristic techniques to remove unwanted knowledge while maintaining model performance (Liu et al., 2022; Maini et al., 2024; Yao et al., 2024; Rafailov et al., 2024). However, these approximate methods lack formal forgetting guarantees and rely on empirical evidence, thus failing to meet the RTBF as stated in applicable legal frameworks.

In this work, we present DP2Unlearning, a novel framework for formal forgetting with guarantees that uses ϵ-differential privacy (DP) (Dwork et al., 2006) on a strategically modified training pipeline to make unlearning *easier*, *cheaper*, and *guaranteed*. Our method involves pre-training LLMs on textual data protected with ϵ-DP, which later enables efficient unlearning of specific data points with the guarantees against disclosure derived from the chosen ϵ parameter. This approach allows LLMs to learn generalizable patterns from the protected data without capturing sample-specific details, which facilitates efficient unlearning through fine-tuning of the retained data.

We demonstrate by means of extensive experiments that DP2Unlearning achieves a similar forgetting and preservation of performance (model utility) to exact unlearning by retraining from scratch, while reducing unlearning costs by nearly half. We also show that DP2Unlearning performs much

better than the existing approximate unlearning methods in both preserving the utility of the model post-unlearning and effectively forgetting the targeted data.

The remainder of this paper is organized as follows. Section 2 discusses related work on unlearning in LLMs. Section 3 provides background on DP. Section 4 presents our DP2Unlearning framework. Section 5 describes the experimental setup. Section 6 reports the experimental results and provides extensive comparisons with the baseline methods. Conclusions and future directions are collected in Section 7. The appendices provide additional experimental results.

2. Related works

We briefly review the literature on exact and approximate unlearning.

2.1. Exact unlearning

Exact unlearning methods, while offering guarantees for complete data removal, are computationally expensive. The simplest approach is to retrain from scratch on the data to be retained; however, this becomes impractical for LLMs due to the high costs involved. Some works focus on more efficient methods for exact unlearning.

SISA (Bourtoule et al., 2021) is a generic exact unlearning framework that can be applied to a variety of ML tasks and models, including LLMs. SISA makes retraining less expensive by sharding training data and slicing shards. Model training checkpoints are recorded after each slice, allowing more efficient retraining from the checkpoints corresponding to the slices containing the forget data. Although it provides exact forgetting guarantees, SISA is not practical for LLMs due to the high computational and memory costs associated with model saving, checking points, retraining, and inference (Blanco-Justicia et al., 2025). Also, while increasing the number of shards reduces unlearning costs, it raises training and inference costs –a different model is trained for each shard, and inference involves an ensemble decision based on the outputs of all shard-level models– and decreases the preservation of the ensemble-level model performance due to increased heterogeneity of the shard-level models.

Adapter Partition and Aggregation (APA) is an exact unlearning technique for the LLM recommendation system (LLMRec) that preserves the inference speed intact (Hu et al., 2024). It works by dividing the training

data into disjoint shards and retraining only the adapters that contain the information that will be forgotten. APA reduces the cost of retraining, but suffers from poor generalizability, insufficient scalability for large data sets, and high memory consumption.

2.2. Approximate unlearning

Approximate unlearning aims to adjust a trained model to eliminate specific knowledge without the need for retraining. Although these methods may not ensure complete formal forgetting, they offer a more computationally feasible alternative. In the following, we review popular approximate unlearning methods that will also serve as baselines for comparison in our experiments.

Gradient ascent (GA) adjusts the parameters of LLM to increase the loss associated with specific data, making it less probable for the model to retain and reproduce that information. Unlike traditional retraining on data to be retained, which focuses on minimizing loss to enhance learning, GA tends to suppress unwanted information by maximizing the loss associated with it. Very recently, several researchers (Liu et al., 2025; Maini et al., 2024; Yao et al., 2024) have adopted GA for approximate unlearning in LLM. Although GA is an efficient alternative to retraining, challenges such as catastrophic forgetting –a situation in which the model unintentionally loses important shared knowledge from the retained set while attempting to forget unwanted data from the forget set– require hybrid approaches that balance forgetting and retention.

Gradient Difference (GD) is a hybrid approach that takes advantage of both gradient ascent and gradient descent (Liu et al., 2022). Unlike gradient ascent, which only increases the error in unwanted information, gradient difference reduces the difference between the error in the data we want to retain and the error in the data we want to forget. At the same time, it keeps the model performing well on the data we want to keep. Recently, several researchers (Maini et al., 2024; Lev and Wilson, 2024; Trippa et al., 2024; Yao et al., 2024) have applied GD for LLM unlearning. While GD's structured optimization makes it a promising alternative to retraining, optimizing its trade-off between forgetting and model performance remains an ongoing challenge.

Kullback-Leibler (KL) minimization aims to minimize the Kullback-Leibler divergence (Hershey and Olsen, 2007) between the probability distributions of the model predictions in the retained data before and after

unlearning. By maintaining the output distributions, KL-based unlearning ensures that retained knowledge stays constant while unwanted information is gradually suppressed. Due to its efficiency in probabilistic alignment, KL divergence minimization has recently been used for LLM unlearning (Yao et al., 2024; Wu et al., 2025; Maini et al., 2024). However, although it provides a less resource-intensive option compared to retraining, it needs additional adjustments to avoid inadvertent loss of knowledge.

Preference Optimization (PO) draws inspiration from direct preference optimization (Rafailov et al., 2024). The idea is to modify the model so that it refrains from generating unwanted information. Alternative answers are generated for the questions that refer to the data to be forgotten (*e.g.*, "I do not know the answer"). Then, the modified model is fine-tuned to minimize a sum of the loss of these alternative answers for the data to be forgotten and the loss of the correct answers for the data to be retained. Recent work has applied PO (Tian et al., 2024; Maini et al., 2024) to adjust LLM outputs for desired behavior after unlearning specific information or patterns. Although effective in some cases, PO faces challenges in balancing the two losses considered.

3. Background on differential privacy

Differential privacy (DP) (Dwork et al., 2006) is a privacy model that ensures that the outputs $\mathcal{A}(\mathcal{D})$ and $\mathcal{A}(\mathcal{D}')$ of a mechanism $\mathcal{A}$ calculated on two data sets, $\mathcal{D}$ and $\mathcal{D}'$, which differ by only one individual's record, remain statistically indistinguishable up to an exponential factor of a parameter ϵ. The formal requirement to achieve pure ϵ-DP is expressed as

$$P[\mathcal{A}(\mathcal{D}) \in \mathcal{R}] \leq e^{\epsilon}\ P[\mathcal{A}(\mathcal{D}') \in \mathcal{R}]$$

In this inequality, $\mathcal{R}$ is a subset of possible output responses that satisfies ϵ-DP and ϵ is called privacy budget, which controls the level of disclosure protection.

The inventors of DP suggest that, for meaningful privacy guarantees against disclosure, the privacy budget (ϵ) should not exceed 1 (Dwork et al., 2019); and that values of ϵ larger than 10 are too weak to provide effective disclosure protection.

The original definition of ϵ-DP has been extended to (ϵ, δ)-DP by including an additive term probability of privacy failure (δ), where $\delta < 1/|\mathcal{D}|$. To achieve indistinguishability, DP typically adds calibrated noise to its output.

DP has some interesting properties:

1. *Immunity to post-processing:* If a mechanism $\mathcal{A}$ satisfies ϵ-DP or (ϵ, δ)-DP, then any post-processing function $g(\cdot)$ applied to its output also satisfies ϵ-DP or (ϵ, δ)-DP, respectively.
2. *Sequential composition:* If a mechanism $\mathcal{A}_1$ satisfies ϵ_1-DP, resp. (ϵ_1, δ_1)-DP, and the mechanism $\mathcal{A}_2$ satisfies ϵ_2-DP, resp. (ϵ_2, δ_2)-DP, then their combined application $\mathcal{A}_{\text{sequential}}$ on the same data set or on non-disjoint data sets satisfies $(\epsilon_1 + \epsilon_2)$-DP, resp. $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$-DP.
3. *Parallel composition:* If mechanisms $\mathcal{A}_1$ and $\mathcal{A}_2$ both satisfy ϵ-DP, resp. (ϵ, δ)-DP, and operate on disjoint data sets $\mathcal{D}_1$ and $\mathcal{D}_2$, then their combined mechanism $\mathcal{A}_{\text{parallel}}$ satisfies ϵ-DP, resp. (ϵ, δ)-DP.

3.1. DP and disclosure protection in LLMs

DP was originally designed to protect queries to structured databases (Dwork et al., 2006). However, DP can also be used to prevent disclosure in language models. DP-MLM (Differentially Private Text Rewriting Using Masked Language Models) and DP-SGD (Differentially Private Stochastic Gradient Descent) are two key mechanisms that we leverage in our work.

3.1.1. DP-MLM

DP-MLM (Meisenbacher et al., 2024) enforces DP on the textual training data. In a privacy-oriented context, DP-MLM should be applied to the noun phrases of each of the documents in the training data set, as they are the most informative units of text –without which it is not possible to disclose specific facts (*e.g.*, private information) about the subjects of the data (Sánchez and Batet, 2016)–.

DP-MLM can be enforced by applying the exponential mechanism, which probabilistically substitutes disclosive terms with semantically similar alternatives while still retaining the general structure of the document. A utility function $u(w, w')$ evaluates the semantic similarity between the original term w and a possible replacement w', which can be calculated based on contextual embeddings and cosine similarity. Specifically, the probability $P(w'|w)$ of swapping w for w' is represented as:

$$P(w'|w) = \frac{\exp(u(w, w')\epsilon)}{\sum_{w'' \in V} \exp(u(w, w'')\epsilon)}$$

where V denotes the vocabulary (the candidate group of words used for substitution).

3.1.2. DP-SGD

DP-SGD (Abadi et al., 2016) and its variant (Kerrigan et al., 2020) are an optimization algorithm that enforces DP during model training by modifying the conventional stochastic gradient descent (SGD) through *gradient clipping*, *Gaussian noise injection*, and *gradient update*. These processes ensure that each data point contributes in a limited and randomized way and prevents the model from learning disclosure information.

1. *Gradient clipping* restricts the impact of a single data item on model training. The gradient update to a specific parameter $\nabla\theta$ is clipped using a set value called the clipping norm C, which is the maximum allowed value (threshold). The clipped gradient is calculated as

$$\bar{\nabla}\theta = \nabla\theta \cdot \min\left(1, \frac{C}{\|\nabla\theta\|_2}\right),$$

   where $\|\nabla\theta\|_2$ represents the L_2-norm of the gradient. This process guarantees that no individual data point excessively affects the optimization procedure, thereby safeguarding against information disclosure.
2. *Gaussian noise injection* consists of adding Gaussian noise $\mathcal{N}_r(0, \sigma^2)$ once the clipping is complete. Noise is added to the combined mini-batch gradient to further obfuscate the impact of specific data pieces before executing the update. The noisy gradient update $\tilde{\nabla}\theta$ that adheres to DP standards is described as

$$\tilde{\nabla}\theta = \sum_{j=1}^{bs} \bar{\nabla}\theta_j + \mathcal{N}_r(0, \sigma^2 I),$$

   where bs is the mini-batch size and σ^2 is the variance of Gaussian noise, adjusted as $\sigma^2 = \frac{C^2 \log(1.25/\delta)}{\epsilon^2}$. This ensures that the algorithm satisfies DP protection with privacy budget ϵ.
3. *Gradient update* is the final process that uses the noisy gradient $\tilde{\nabla}\theta$, the learning rate η, and the model parameter at the t-th iteration θ_t. Thus, the resulting gradient update rule is $\theta_{t+1} = \theta_t - \eta\tilde{\nabla}\theta$. This update ensures that no single data point substantially impacts the training of the model.

4. DP2Unlearning

The exact unlearning methods aim to completely eliminate the data that must be forgotten. However, in order to comply with privacy and copyright laws, complete data removal is overkill. For example, the GDPR states that it is sufficient to make personal data non-personal (for example, through anonymization) to be beyond the regulation's scope. Similarly, for copyright protection, it is enough to prevent verbatim reproduction of the original source while still preserving the underlying semantics.

This means that, in practice, what we need is *selective but guaranteed* unlearning. That is, the model should exactly forget specific details while still being able to retain the general meaning. Our hypothesis is that privacy models such as *differential privacy* (Dwork et al., 2006), *k-anonymity* (Sweeney, 2002), or their variants, can be used to enforce on the trained model outputs this selective or partial removal of training data with guarantees against (detailed) information disclosure.

Given the heterogeneity and lack of structure of the textual documents employed to train LLMs, DP seems the best-suited model for this task, as it allows for enforcing disclosure protection on documents individually and independently. This is particularly beneficial when scaling to large data sets involved in training LLMs.

On the other hand, ϵ-DP offers *ex ante* guarantees against disclosure, which ensure that no specific data point can be distinguished from other points based on the model output up to an exponential factor depending on ϵ. In practice, this guarantees that the model is guaranteed up to that exponential factor not to reproduce any private or copyright-protected information on which DP has been applied. The actual degree of protection against disclosure can be controlled by the privacy parameter ϵ, which in our case dictates the level of forgetting. For example, larger values of ϵ (*e.g.*, $\epsilon > 10$) offer mild forgetting, which would tend to approximate forgetting, while $\epsilon = 0$ is equivalent to complete exact forgetting. Middle-ground values would probably be the best suited to provide guaranteed but utility-preserving forgetting.

Using the intuitions above, we propose DP2Unlearning, an LLM construction framework that uses DP with a modified training pipeline to make unlearning *cheaper* and *guaranteed*. The framework is designed in such a way that it can handle forgetting requests efficiently while adhering to privacy and copyright regulations.

4.1. Method description

DP2Unlearning operates in three stages: (A) Unlearning-ready training, (B) Pre-unlearning fine-tuning, and (C) Unlearning execution. The framework executes the first two stages, (A) and (B), only once, while the final stage, (C), repeats for each unlearning request. The workflow of DP2Unlearning is depicted in Figure 1.

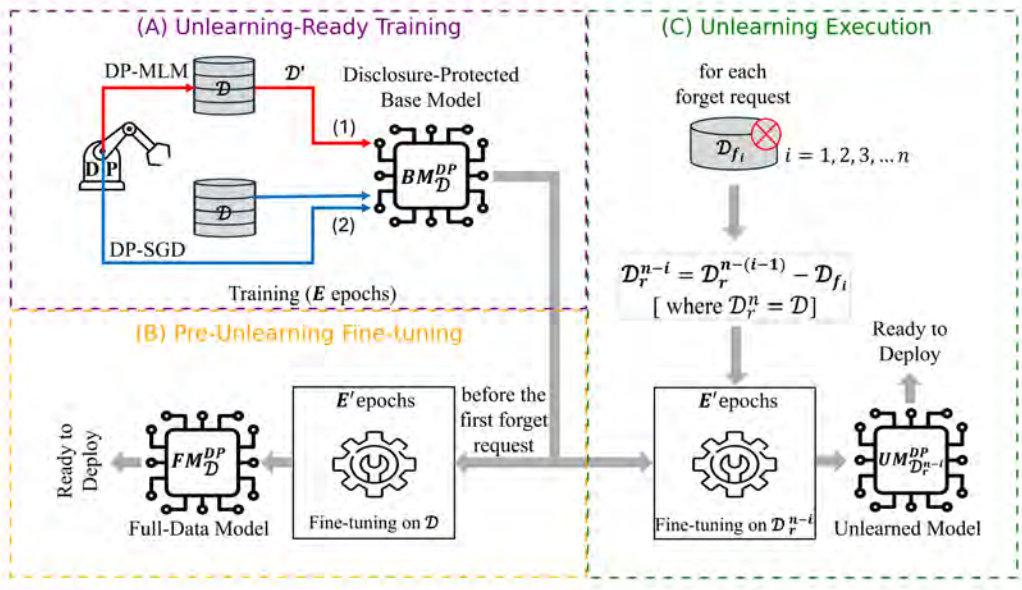


Figure 1: Workflow of the proposed DP2Unlearning framework

Stage (A) –**unlearning-ready training**– involves training a disclosure-protected base model (or simply a base model), ensuring safeguards against the disclosure of any information that may need to be unlearned. As anticipated in Section 3, this can be accomplished in two different ways:

1. DP-MLM-based data protection: The model is trained on a data set ($\mathcal{D}'$) where the specific text components are obfuscated using DP-MLM. Protection occurs at the individual data point level by substituting disclosure terms, mostly noun phrases, with semantically similar alternatives in a probabilistic manner through the use of an exponential mechanism. The choice of ϵ directly governs the disclosure guarantee, with smaller values ensuring stronger protection. However, depending on the trade-off between disclosure protection and model utility, a relaxed

version (ϵ, δ)-DP can be implemented, introducing limited uncertainty in token selection. Since DP-MLM functions at the document level, the loss of privacy accumulates across multiple token substitutions, and the total protection adheres to the sequential composition theorem.

2. DP-SGD-based training: DP-SGD directly imposes (ϵ, δ)-DP constraints during training by adding Gaussian noise to gradient updates, ensuring that individual contributions remain indistinguishable. Although DP-SGD uses ϵ to control the level of disclosure protection, it also requires a small value δ, since pure ϵ-DP is theoretically impossible with Gaussian noise (with Gaussian noise, there is always a small chance that some updates may exceed the pure ϵ-DP limit, which requires δ to be taken into account).

In both cases, the model is trained until convergence for E epochs, resulting in the base model ($BM_{\mathcal{D}}^{\mathrm{DP}}$), which we must preserve indefinitely because unlearning rests on it. The development of the base model is resource intensive, although it is performed only once. In this respect, an advantage of DP-MLM over DP-SGD is that in the former DP protection does not need to be applied to all training data: public domain training data sources (*e.g.*, Wikipedia), for which forgetting requests will never apply, can be kept and used 'as is', significantly reducing training costs. In contrast, DP-SGD treats all data equally.

In stage (B) –**pre-unlearning fine-tuning**– one tries to make up for the decrease in model performance that is likely to have occurred as a result of DP-protection in stage (A): the DP-protected base model is expected to exhibit lower model performance compared to a model trained without DP. We hypothesize that by fine-tuning this safeguarded model on the original raw data ($\mathcal{D}$), we can restore most (if not all) of its performance; also, due to the incremental nature of fine-tuning and the fact that it is done on the same data sources used for training (even though, in this case, unprotected), it will require significantly less computational resources than retraining the base model on $\mathcal{D}$ from scratch. This should be possible because initial DP-induced training, while not producing an accurate model, allows it to gain a "general" understanding of $\mathcal{D}$, thus facilitating faster learning of the details and specifics of $\mathcal{D}$ through fine-tuning. As a result, and as long as there is no forget request, we release a version of $BM_{\mathcal{D}}^{\mathrm{DP}}$ fine-tuned on the complete raw data ($\mathcal{D}$) for E' epochs, which corresponds to the initial full-data model ($FM_{\mathcal{D}}^{\mathrm{DP}}$) ready for deployment. Note that this fine-tuning step uses raw un-

protected data to restore the utility lost due to DP protection and, therefore, does not provide DP guarantees. The resulting model is deployed only until the first unlearning request is received. Upon such a request, the system reverts to the DP-protected base model and fine-tunes it on the retain set to offer disclosure protection guarantees on the data to be forgotten.

Stage (C) –**unlearning execution**– is run whenever a forget request $\mathcal{D}_{f_i}$ arrives, where $i = 1, 2, 3 \ldots, n$. Specifically, the current deployed model is discarded and the saved base model $BM_{\mathcal{D}}^{\text{DP}}$ is resumed. The latter model is then fine-tuned again using only the data set to be retained, which is

$$\mathcal{D}_r^{n-i} = \mathcal{D}_r^{n-(i-1)} - \mathcal{D}_{f_i}, \quad \text{where} \quad \mathcal{D}_r^n = \mathcal{D}.$$

This results in an unlearned model ($UM_{\mathcal{D}_r^{n-i}}$) with an expected performance similar to retraining from scratch on the retained data, but with ϵ-DP or (ϵ, δ)-DP guarantees against the disclosure of the data to be forgotten ($\mathcal{D}_{f_i}$). In other words, we achieve DP disclosure guarantees on data to be forgotten because the original data points in $\mathcal{D}_{f_i}$ have not been seen by the unlearned fine-tuned model (they have only been seen by the base model under DP protection):

$$UM_{\mathcal{D}_r^{n-i}} = \text{Fine-tuning}(BM_{\mathcal{D}}^{\text{DP}}, \mathcal{D}_r^{n-i}, E').$$

4.2. Unlearning guarantees and computational cost

We prove that either ϵ-DP or (ϵ, δ)-DP guarantees against disclosure are satisfied for a forgetting request.

Proposition 1 (Disclosure). *Unlearning with stage (C) fulfills the ϵ-DP requirement when utilizing DP-MLM and the (ϵ, δ)-DP requirement when utilizing DP-SGD.*

Proof: Based on the post-processing immunity characteristic of DP (refer to DP characteristics in Section 1), the ϵ-DP or the (ϵ, δ)-DP guarantee extend to $UM_{\mathcal{D}_r^{n-1}}^{\text{DP}}$, ensuring that $\mathcal{D}_{f_i}$ remains protected against disclosure proportionally to the chosen ϵ. □

We now analyze the computational cost of the above three stages:

- Stages (A) and (B) together *incur a large computational cost, but only once*. As mentioned earlier, stage (A) is resource intensive, primarily

due to the added computational burden of DP protection. For DP-SGD, the associated cost is higher than that of DP-MLM due to the following key factors: (i) per-sample gradient computation and clipping, which adds a computational overhead during training by limiting gradient values to avoid substantial updates that may compromise disclosure information, and (ii) noise injection, which introduces randomness and delays convergence. Consequently, DP-SGD typically incurs longer convergence times than DP-MLM, which uses probabilistic term substitution to obfuscate disclosure information. Moreover, DP-SGD needs to process all data during training, whereas DP-MLM can be used on a sensitive subset (*e.g.*, private or copyrighted data). Stage (B) also introduces a one-time overhead, as it involves fine-tuning the base model with unprotected data to recover model performance for deployment; this is a process that is not required in standard LLM training.

- Stage (C) fine-tunes an already trained model (the base model) rather than starting from scratch. This fine-tuning benefits from the knowledge embedded in the base model. Thus, fewer optimization steps are required to recover the performance lost due to DP protection. Therefore, processing a forgetting/unlearning request with stage (C) requires significantly less computational cost than retraining from scratch. Specifically, stage (C) reduces the cost by a factor of E'/E, where E is the number of epochs required to train the model from scratch on the data to be retained, and E' is the number of epochs needed to fine-tune the protected base model on the data to be retained.

5. Experimental setup

5.1. Data sets and models

Due to the high computational training costs of LLMs, we cannot afford to train a full-fledged LLM from scratch. Instead, we use pre-trained models –*Phi-1.5B* (from Microsoft) and *Llama2-7B* (from Meta)– of varying sizes and capabilities. To have control over the data to be unlearned, we further train these models using additional data sets specifically tailored for evaluating unlearning.

As additional training data, we used the TOFU (Maini et al., 2024) data set, a recent benchmark data set specifically designed to evaluate unlearning

in LLM. The data set includes 4,000 question-answer pairs derived from 200 varied synthetic author profiles, each comprising 20 question-answer pairs. The data set is synthetically created and intentionally modified to ensure that it does not overlap with the training data typically used to build an LLM. This intentional design makes the TOFU data set a versatile resource for a controlled and unbiased evaluation of unlearning methodologies.

The TOFU data set is divided into Forget and Retain sets, with varying proportions of 1%-99%, 5%-95%, and 10%-90%, allowing investigation of the impact of different unlearning ratios. Additionally, TOFU includes two supplementary real-world data sets: Real Authors (containing real author-related questions) and Real-World Facts (covering general knowledge questions).

Since TOFU data were intentionally created to ensure that they were not used to pre-train existing LLMs, further training a pre-trained LLM on TOFU allows experimenting with unlearning, because training the pre-trained LLM on the retain subset of TOFU can be viewed as retraining an LLM from scratch on the retain subset. We name this setting *retraining from scratch on Retain set (RFS-R)*, which corresponds to *exact unlearning*.

5.2. Baseline methods

To compare our approach with those of related work, we reproduced several approximate unlearning methods. Specifically, we use the Gradient Ascent (GA), Gradient Difference (GD), Kullback-Leibler Minimization (KL) and Preference Optimization (PO) methods introduced in Subsection 2.2. Each method adopts a different strategy:

- GA reverses the model updates by maximizing the loss function, thus compelling the model to unlearn previously acquired (unwanted) knowledge.
- GD modifies gradients to reduce model retention of the information intended to be forgotten while preserving its usefulness.
- KL forces the model output to match a reference distribution by minimizing the KL divergence.
- PO utilizes direct preference optimization to modify the model's predictions away from data considered to be forgotten.

We implemented these baselines using the TOFU unlearning implementation, which is publicly available at `https://github.com/locuslab/tofu`.

5.3. Evaluation metrics

Our evaluation primarily focuses on model utility (*i.e.*, the model's ability to retain useful knowledge post-unlearning) and forget quality (*i.e.*, the extent to which the model effectively forgets unwanted knowledge). More specifically, model utility refers to the model's ability to provide correct responses for information it is supposed to retain, ensuring that unlearning does not harm the model's overall performance. On the other hand, the forget quality indicates how effectively the model stops giving accurate responses to the information it is supposed to forget. Note that model utility can also be used to evaluate forget quality: high post-unlearning utility with respect to the data to be forgotten can be viewed as an indication of poor forget quality.

To evaluate model utility, we used the evaluation metrics –ROUGE-L, conditional probability, and truth ratio– used in TOFU (Maini et al., 2024).

- *ROUGE* estimates the similarity between the responses generated by the model and the ground truth answers (Lin, 2004), allowing for minor differences in wording. ROUGE-L evaluates similarity by computing the longest common subsequences (LCS) of words between the model-generated response and the correct (ground truth) answer. This metric gives a score that shows how accurate the content is; even if the words are not exactly the same, it provides a high score for semantically equivalent content. A typical way to calculate ROUGE-L recall is

$$\text{ROUGE-L} = \frac{\text{LCS}(GTT, MGT)}{|GTT|},$$

  where GTT is the ground truth text (reference text), MGT is the model generated text (hypothesis), $\text{LCS}(GTT, MGT)$ represents the length of the longest common subsequence between GTT and MGT, and $|GTT|$ is the total number of tokens in the reference text.

- *Conditional probability* measures the confidence of the model in its predictions. It helps in assessing retention effectiveness. The formula used for a query and response pair (Q, r) in the Forget Set and the Retain Set is

$$P(r|Q)^{\frac{1}{|r|}},$$

  where $P(r|Q)$ is the probability that the model returns the response r when asked query Q, normalized by the length $|r|$ of the response

(as done in (Cho et al., 2014)). However, in the Real Authors and Real World Facts data sets, the conditional probability is computed for multiple choice questions as

$$\frac{P(r_1|Q)}{\sum_{i=1}^{x} P(r_i|Q)},$$

where r_1 represents the correct answer among the x options. This process makes it easier to compare answers of different lengths.

- *Truth ratio* (TR) measures how well a model prioritizes a correct response over multiple incorrect responses (Lin et al., 2022), reflecting its retained knowledge. Mathematically, the TR score is calculated by dividing the average likelihood of intentionally modified incorrect responses (that is, responses following the same linguistic pattern as the right answer but including inaccuracies that sound believable, yet are incorrect) by the likelihood of a correct paraphrased response (*i.e.*, a semantically accurate rewording of the original answer, making sure that the different wording does not change the meaning). Therefore, this metric quantifies the extent to which the model, even after unlearning the unwanted knowledge, continues to prioritize providing correct responses rather than incorrect ones:

$$TR = \frac{\frac{1}{|R_{\text{inaccurate}}|} \sum_{\hat{r} \in R_{\text{inaccurate}}} P(\hat{r}|Q)^{\frac{1}{|\hat{r}|}}}{P(\tilde{r}|Q)^{\frac{1}{|\tilde{r}|}}},$$

where $R_{\text{inaccurate}}$ is the set of intentionally modified responses designed to be incorrect, $\tilde{r}$ is the accurate paraphrased response, and $\hat{r}$ is an intentionally modified incorrect response.

Overall model utility: To measure the utility of the model as a whole, we evaluated the above three metrics in three data sets: Retain Set, Real-World Facts, and Real Authors. We normalized each of the three evaluation metrics to fall within the range $[0, 1]$, where higher values indicate improved retention (utility preservation). To consolidate these metrics into a single model utility, we calculated the harmonic mean of the nine metric values (three values from each of the three data sets). Since the harmonic mean is sensitive to low values, a significantly low score of any of the nine evaluation metrics will disproportionately lower the overall model utility score.

Overall Forget Quality: To measure forget quality as a whole, we assessed the above three metrics on the forget data set. Due to the intricate nature of LLMs, the evaluation of forgetting quality often relies on statistical tests and changes in data distribution (Goel et al., 2022). Specifically, we leverage the Kolmogorov-Smirnov (KS) test, a non-parametric statistical test to assess the variations in truth ratios between the unlearned model and a model trained solely on retained data (RFS-R) to check how well a model can forget information. Basically, the KS test evaluates two cumulative distribution functions (CDFs) and computes (i) the KS statistic (D_{KS}): the maximum absolute difference between the two CDFs; and (ii) the p-value, that is, the probability that the two samples come from the same distribution, which *indicates the forget quality*. To find how different the two CDFs are, the KS statistic (D_{KS}) can be calculated as

$$D_{KS} = \max |C_{\mathrm{u}}(x) - C_{\mathrm{r}}(x)|,$$

where $C_{\mathrm{u}}(x)$ and $C_{\mathrm{r}}(x)$ are the CDFs of truth ratios for unlearned and retain-only models (RFS-R, the benchmark for comparison), respectively. A higher value D_{KS} implies that the CDFs are statistically different and therefore indicates unsuccessful unlearning. On the other hand, a lower value D_{KS} implies that CDFs are statistically identical and therefore indicate successful unlearning. We used a 0.05 threshold for the probability p-value of the KS test: a value below 0.05 clearly allows us to reject the null hypothesis that the two CDFs are statistically the same, indicating ineffective forgetting; while a high p-value ($p \geq 0.05$) indicates that the unlearned model closely follows the retain-only model, indicating good forgetting. The threshold 0.05 for the p-value of the KS test is a widely accepted standard in the testing of statistical hypotheses (Aslam, 2019; Maini et al., 2024). However, its justification depends on the context of unlearning. Although smaller p values (*e.g.*, $p < 0.01$) are sometimes preferred in high-sensitivity applications to reduce false positives, in practical unlearning scenarios, a threshold of 0.05 remains a standard choice. Given our data set size of 4,000 instances, this threshold is justified, as it avoids excessive sensitivity to minor distributional shifts while still detecting meaningful differences in forget quality.

5.4. Training settings

We next detail the technical configuration and training parameters used in our experiments to ensure reproducibility and facilitate independent valida-

tion of our results. We cover the training settings for unlearning-ready training, pre-unlearning fine-tuning, unlearning execution, DP configurations, and baseline methods unlearning. The code of our experiments is available at https://github.com/tamimalmahmud/DP2unlearning/

Unlearning-ready training: To determine the optimal number of training epochs for the base model, we analyzed model convergence without DP training. This established a reference point for selecting DP-aware training epochs.

We empirically found that both the *Phi* and *Llama2* models reached a near-perfect ROUGE-L score of ≈ 1.0 when they were trained on TOFU full data without DP for the $E = 10$ and $E = 6$ epochs, respectively, as shown in Figure 2 and Table 1.

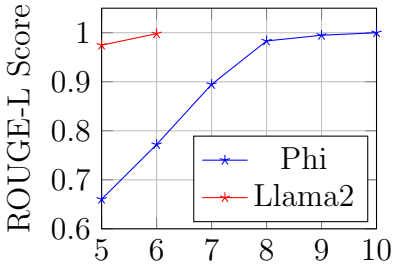


Figure 2: ROUGE-L at different epochs

Table 1: ROUGE-L at different stages. For *Phi*, $E = 10$ and $E' = 5$. For *Llama2*, $E = 6$ and $E' = 3$.

Model	Pre Trained	Trained without DP E epochs	Our $BM_{\mathcal{D}}^{\mathrm{DP}}$ E epochs	Our $FM_{\mathcal{D}}^{\mathrm{DP}}$ E' epochs
Phi	0.4494	1.00	0.4233	0.9957
Llama2	0.3549	0.9972	0.3834	0.9789

This observation guided our choice: we aligned the DP-aware training epochs with the non-DP training convergence points. We set $E = 10$ for *Phi* (trained with DP-MLM and DP-SGD) and $E = 6$ for *Llama2* (trained only with DP-MLM, as DP-SGD is computationally prohibitive for our setup).

Pre-unlearning fine-tuning: We empirically determined that fine-tuning for the $E' \approx E/2 \approx 5$ epochs (for *Phi*) and the $E' \approx E/2 \approx 3$ epochs (for *Llama2*) significantly improved their ROUGE-L scores, bringing them close to 1.00 (see Table 1).

Unlearning execution: Since unlearning is performed using the same fine-tuning approach as in pre-unlearning fine-tuning, we adopt the same convergence point. Therefore, to process each forget request, we fine-tune the corresponding base model ($BM_{\mathcal{D}}^{\mathrm{DP}}$) exclusively on the retain data for $E' \approx E/2 \approx 5$ epochs (for *Phi*) and $E' \approx E/2 \approx 3$ (for *Llama2*), respectively.

Privacy settings for DP-MLM and DP-SGD: We experimented with different values of the privacy budget ($\epsilon = 0.5, 1, 10, 25, 100$) to find the best balance among disclosure protection guarantees, model performance and

computational overhead. Additionally, we optimized the configurations for each DP mechanism as follows:

- for DP-MLM, we set the logit clipping bounds to clip_min = -5.2093 and clip_max = 20.3048 to ensure controlled sensitivity when selecting substitute tokens, which we found to provide the best trade-off between disclosure protection and semantic coherence.

- for DP-SGD, we set the minimum possible δ to make it nearly equivalent to pure ϵ-DP, as DP-SGD inherently requires $\delta > 0$ as discussed in the methodology. Since achieving pure ϵ-DP ($\delta = 0$) is theoretically impossible, we followed the best practice of setting (δ to be smaller than $1/|\mathcal{D}|$, see Section 3 for justification), and ensured $\delta \ll 2.5 \times 10^{-4}$ given the 4,000 instances of the data set.

Baseline methods: To ensure a fair comparison that aligns with our unlearning execution settings, we applied the same fine-tuning settings across all baseline methods. We set $E' = 5$ epochs for Phi and $E' = 3$ epochs for Llama2.

Hardware setup: All experiments were performed on an NVIDIA H100 GPU with 80 GB of HBM3 memory. We used a learning rate 5×10^{-5}, a weight decay 0.01, and an effective batch size 16 (batch size 4, gradient accumulation steps 4).

6. Experimental results

We now proceed to evaluate the two variants of our DP2Unlearning framework (i) **DP2U-SGD**, when the model is trained with DP-SGD and (ii) **DP2U-MLM** when the model is trained on data protected by DP-MLM.

We first analyze the trade-off between disclosure protection and model performance. This is crucial since an increased privacy budget can compromise disclosure protection and thus unlearning guarantees, while a reduced budget degrades the model utility, thereby requiring more fine-tuning effort to recover utility. To identify the best balance, we systematically evaluated key performance metrics at various values of ϵ.

Subsequently, we performed a comparative analysis to evaluate the effectiveness of our approaches by comparing them with exact unlearning through RFS-R and several approximate unlearning baselines (discussed in Section 2.2).

6.1. Balancing disclosure protection and model performance

We analyze how varying the values of ϵ (*i.e.*, $0.5, 1, 10, 25, 100$) affected ROUGE, the utility of the model, and the quality of forgetting. Figures 3 and 4 show the results for the Phi and Llama2 models, respectively, across three model states: the base model (trained with DP), the full data model (fine-tuned with full original data) and the unlearned model (fine-tuned only on the data to be retained).

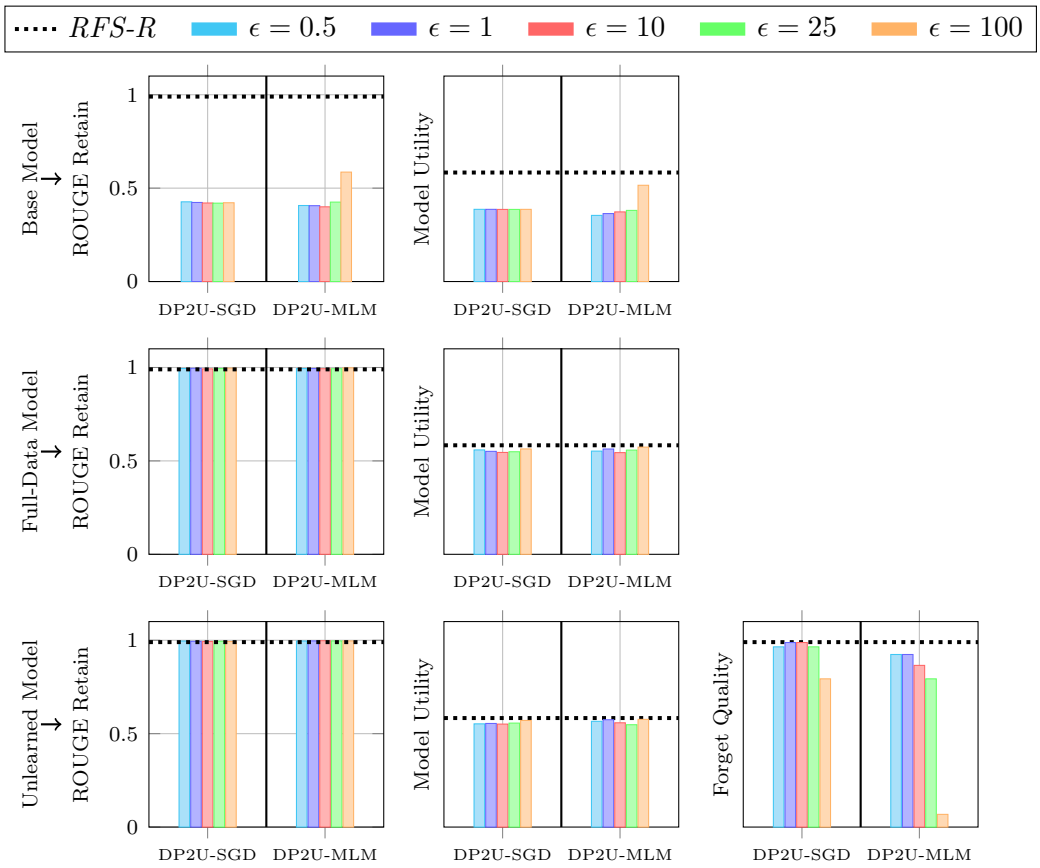


Figure 3: Evaluation results for Phi and 5% forget ratio. $E = 10$ for *RFS-R* (Retrain from scratch on the set to be retained) and $E' = 5$ for DP2U-SGD and DP2U-MLM for all ϵ values.

We observe the following.

DP2U-SGD: On Phi (the computational constraints of DP-SGD prevent training Llama2 within our setup), DP2U-SGD maintains consistency

in all metrics regardless of the different values of ϵ. Generally, larger ϵ values enhance model utility by compromising disclosure restrictions, whereas smaller ϵ values provide better disclosure protection, albeit with a reduction in utility. However, consistent results are mainly due to the DP-SGD mechanism of controlled noise addition and gradient clipping, which guarantees that updates are limited and are quite insensitive to values ϵ. As a result, while the ROUGE retain and model utility metrics in the unlearned model improve and closely match those of RFS-R, the values remain nearly identical across all ϵ levels for all model states. This aligns with the theory that DP-SGD protects disclosure information by limiting individual gradient contributions rather than completely removing learned information.

DP2U-MLM: Unlike DP-SGD, DP-MLM relies on replacing disclosive terms (mainly noun phrases) with probabilistically chosen tokens, which makes it fairly affected by ϵ. This effect is particularly pronounced in larger models such as Llama2. As ϵ increases, the utility of the model increases due to fewer token substitutions, allowing the model to retain the semantic structure better. In contrast, at lower ϵ values, excessive token replacements degrade the learned representations, and fine-tuning struggles to fully restore model utility. Interestingly, this pattern is not as pronounced in the smaller Phi model, which shows fairly consistent performance at various values of ϵ. Given that most LLMs are more similar to Llama2 in size and architecture than Phi, these findings imply that the sensitivity observed to ϵ is likely to be relative to a wider range of models. Despite these effects, in the unlearned model, DP-MLM demonstrates enhanced performance in all metrics and brings them close to the benchmark RFS-R. Importantly, while the utility of the model shows minimal change at various values of ϵ, the forget quality shows considerable variation. This suggests that the choice of ϵ critically influences the balance between disclosure protection and model performance.

Choosing ϵ: According to the recommendations of the inventors of DP (mentioned in Section 3), $\epsilon \leq 1$ achieves strong disclosure protection, which is critical for *guaranteed* unlearning in our approaches. However, setting ϵ should balance disclosure protection and model performance. These are our findings based on the results reported above:

- **For $\epsilon < 1$:** The base model reduces the utility of the model (more apparently in the larger model, Llama2) but provides greater disclosure protection after unlearning. This is closer to RFS-R, as it provides

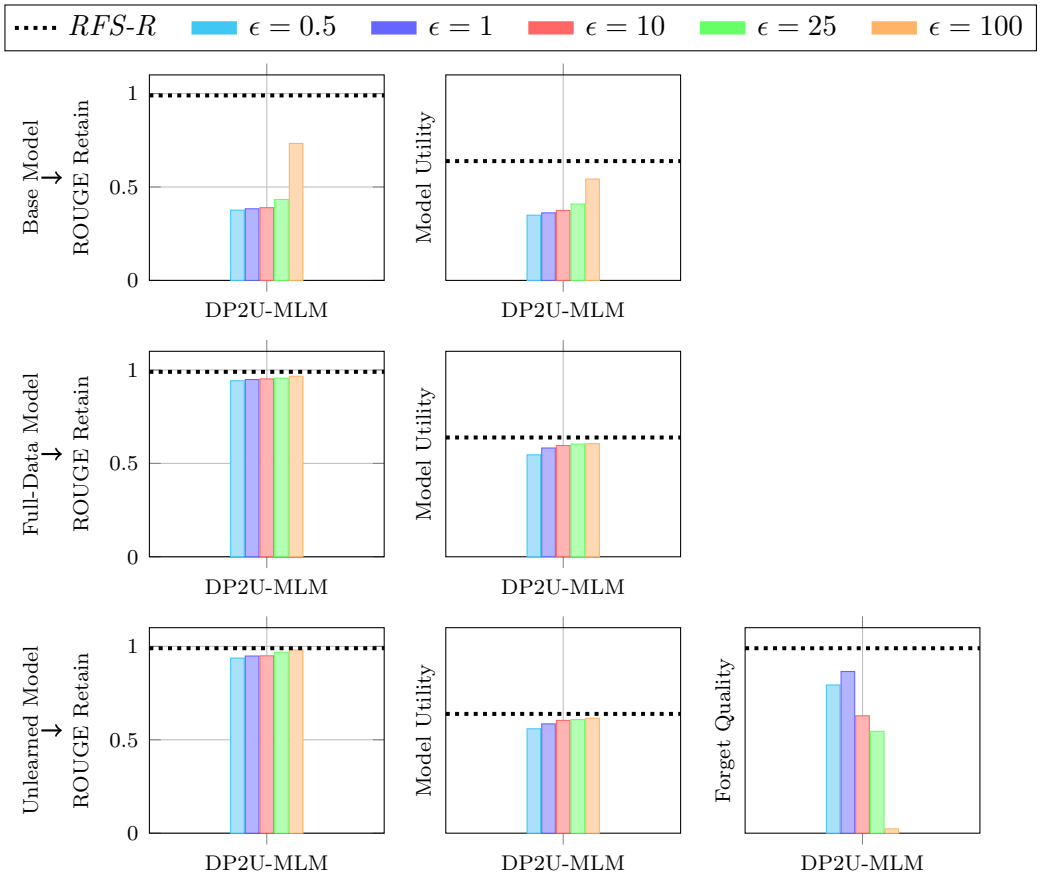


Figure 4: Evaluation results for Llama2 and 5% forget ratio. $E = 10$ for *RFS-R* and $E' = 5$ for DP2U-MLM for all epsilon values.

guaranteed protection against disclosure but requires more extensive fine-tuning to recover the utility.

- **For $\epsilon > 10$:** The base model retains higher model utility (more apparently in the larger model, Llama2), but may compromise disclosure protection after unlearning. This is closer to approximate unlearning, as less effort is required to restore the utility of the model by sacrificing unlearning guarantees.

Therefore, based on our findings, $\epsilon = 1$ offers a good trade-off, as it provides theoretically and practically meaningful protection against disclosure (*i.e.*, guaranteed unlearning) with manageable fine-tuning requirements. This result aligns with the recommendation of the DP inventors.

6.2. Comparative analysis

Next, we compare our approach with exact unlearning (RFS-R) and approximate unlearning techniques (GA, GD, KL, and PO). As a reference, in Table 2 we report the utility of the models before unlearning. The evaluation results for the Retain and Forget sets are detailed in Table 3 and depicted in Figures 6 and 7. Additional results for the four baseline data sets are included in Appendix A.1.

Table 2: Utility of the models before unlearning. FT-RF stands for fine-tuning on both retain and forget datasets.

Models	**Pre-trained**	**Non-DP FT-RF**
Phi	0.3354	0.5411
Llama2	0.2516	0.5793

For a fair comparison, DP2U-MLM and DP2U-SGD were evaluated using the chosen privacy budget $\epsilon = 1$. Additional results for $\epsilon = 0.5, 10, 25, 100$ are presented in Appendix A.4 and Appendix A.5.

6.2.1. Overall forget quality and model utility

Table 3 presents the overall forget quality (FQ) and model utility (MU) across different unlearning methods for the Phi and Llama2 models, evaluated at three forget ratios (1%, 5%, and 10%). These results highlight the trade-offs between retention and unlearning effectiveness in various approaches.

With 1% forgetting, GA, GD, KL, and PO achieve some varying degrees of forget quality and model utility. However, PO does not exceed the 0.05 KS p value threshold, indicating that it does not satisfy statistically significant forgetting. Furthermore, for GA, GD, and KL, the forget quality remains low for Phi, whereas Llama2 exhibits a trade-off between forget quality and model utility, suggesting that model size and architecture influence forgetting performance.

With larger forget ratios (5% and 10%), all approximate methods fail to achieve a meaningful forgetting (KS p-value ≥ 0.05). In particular, PO and GD exhibit some level of model utility but negligible forget quality, implying that they do not effectively remove unwanted information. In general, all approximate methods exhibit extremely poor forget quality at higher forget ratios, often approaching near zero. This highlights that these methods are ineffective for reliable long-term forgetting, especially when more data need to be forgotten.

Table 3: Forget quality and model utility of different unlearning methods for Phi and Llama2 at different forget ratios. For each metric, the best result is **bolded**, and the second-best is underlined. *RFS-R* serves as the benchmark but is excluded from ranking due to its high computational cost. Our methods are highlighted by gray cell color.

Model	Method	Epochs	Forget Ratios					
			1%		5%		10%	
		↓	MU ↑	FQ ↑	MU ↑	FQ ↑	MU ↑	FQ ↑
Phi	RFS-R	10	0.5448	1.0000	0.5380	1.0000	0.5314	1.0000
	GA	5	0.0230	0.0143	0.0000	0.0021	0.0000	8.84E-08
	GD	5	0.4329	0.1650	0.1982	1.87E-09	0.4005	5.56E-14
	KL	5	0.0210	0.0143	0.0000	0.0021	0.0000	1.46E-14
	PO	5	**0.5223**	0.0013	0.5114	2.56E-14	**0.5313**	7.90E-22
	DP2U-SGD	5	0.5060	**0.9900**	0.5122	**0.9878**	0.5113	0.9003
	DP2U-MLM	5	0.5026	**0.9900**	**0.5223**	0.9238	0.5134	**0.9014**
Llama2	RFS-R	6	0.5870	1.0000	0.5711	1.0000	0.5688	1.0000
	GA	3	0.0000	0.7659	0.0000	2.61E-07	0.0000	1.85E-15
	GD	3	0.0000	0.2657	0.3490	1.39E-11	0.4053	2.86E-14
	KL	3	0.0000	0.4046	0.0000	4.61E-07	0.0000	2.59E-12
	PO	3	0.4905	0.0013	0.4950	1.83E-19	0.5290	2.43E-19
	DP2U-MLM	3	**0.5231**	**0.9999**	**0.5320**	**0.8655**	**0.5378**	**0.1761**

Both DP2U-SGD and DP2U-MLM achieved results comparable to exact forgetting RFS-R, showcasing their robustness. Although DP2U-SGD maintains high model utility while ensuring strong forget quality, it requires significant computational resources, making it impractical for large models. In contrast, DP2U-MLM stands out as the best performing method for Phi and Llama2, balancing high utility (MU is 0.5222 for Phi and 0.5323 for Llama2, close to the MU before forgetting reported in Table 2) and strong forgetting quality (FQ is 0.9238 for Phi and 0.8655 for Llama2), even at moderate forget ratios of 5%. These results confirm that DP2U-MLM provides a compelling alternative to RFS-R at a fraction of the computational cost, making it ideal for scalable unlearning in resource-constrained environments.

To gain a more comprehensive understanding of how the performance of the model changes over epochs, Figure 5 presents an in-depth comparison of the forget quality and the model utility for various unlearning techniques at 5% forget ratio. The epoch-wise evolution for the 1% and 10% forget ratios is presented in Appendix A.2 to offer a more thorough perspective on how

the methods perform under different forget conditions.

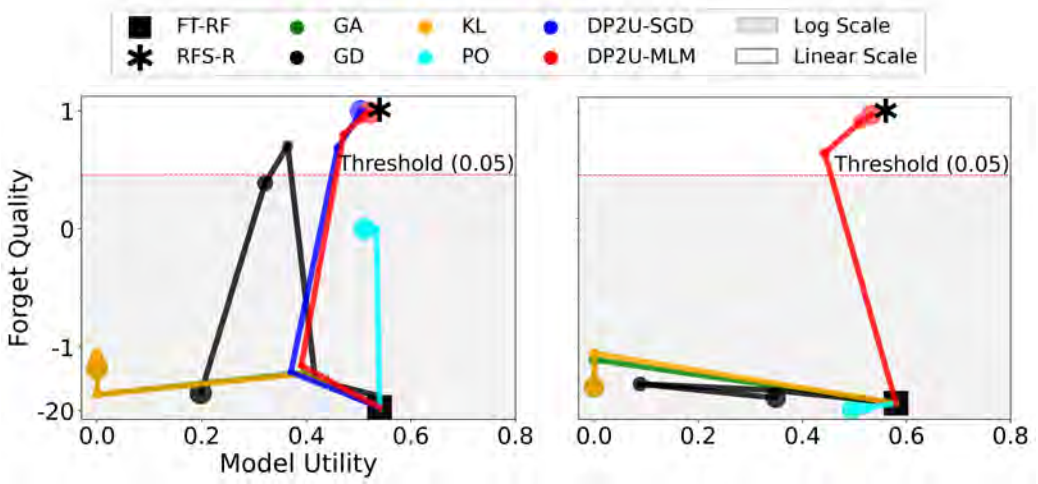


Figure 5: Forget quality vs. model utility for the Phi (left) and Llama2 (right) across different unlearning methods and epochs. The size of markers grows with the number of epochs.

Figure 5 illustrates the trade-off between forget quality and model utility across different unlearning techniques for Phi (left) and Llama2 (right) models. Each method is evaluated at multiple epochs, where the comparative size of the markers (circles, squares, and asterisks) represents the number of epochs. The larger the epoch number, the larger the marker size. The epochs are $E = 10$ for FT-RF and RFS-R, while $E' = 5$ for other methods.

It is evident from the results that, with the exception of GD at some epoch for the Phi model, all approximate techniques failed to achieve the minimum meaningful forgetting quality (*i.e.*, a KS p-value above the threshold of 0.05). Further, the behaviors of different unlearning methods across epochs are far from uniform. Fluctuations are particularly evident in approximate methods, where forget quality varies as epochs increase. These methods also display a clear trade-off with model utility: as the number of unlearning epochs increases, performance declines, causing utility to drop sharply (with trajectories shifting left), often approaching zero. In particular:

- For GD, the fluctuations are related to the nature of gradient-based updates. In early epochs, when the loss is high, GD makes large updates, aggressively unlearning the forget set. However, as loss decreases, the

gradients become smaller and can reverse the direction of updates, unintentionally reinforce forgotten information, and damage retention. This non-monotonic behavior is more pronounced in larger models like Llama2, where the complex loss landscape makes updates less predictable.

- PO exhibits a more consistent trend in terms of model utility, albeit with a reduced forget quality. This happens because PO replaces responses on forgotten data with predetermined responses, which limits parameter adjustments and consequently preserves higher model utility. However, as these predetermined responses differ significantly from those obtained from a model retrained from scratch, PO leads to poor forget quality. This trade-off between forget quality and model utility is more pronounced in Llama2, whose larger parameter space and complexity intensify these problems.

- GA and KL repeatedly demonstrate poor forget quality along with a notable degradation in model utility as the number of epochs increases for both models. This occurs because the loss function is optimized for the forgotten data, which impacts the overall model and results in less effective forgetting than that observed in retrained models.

Previous research (Maini et al., 2024) noted similar inconsistencies in approximate unlearning methods. These findings highlight the limitations of conventional approximate unlearning techniques in achieving reliable forgetting.

In contrast, both the DP2U-SGD and DP2U-MLM approaches exhibit a remarkably positive trend. Although they begin with low model utility and forget quality, they show significant improvement as the number of epochs increases and gradually converge toward the model utility and forget quality obtained by RFS-R, which is the benchmark for exact unlearning. Importantly, both methods maintain this positive trend in all forget ratios (results for 1% and 10% forget ratios are presented in Appendix A.2). This highlights their ability to adapt effectively to varying unlearning requests and model configurations, ensuring consistent performance even in more diverse and challenging contexts. More specifically, *after only two epochs for Phi and one epoch for Llama2, our approach outperforms all approximate unlearning baselines and surpasses the threshold of meaningful forgetting.*

Although Figure 5 illustrates the promising convergence of DP2Unlearning methods with exact unlearning benchmarks, it also reveals some degree of variability in forget quality between epochs and forget ratios. The underlying causes of these variations are discussed in detail in the following paragraphs.

We observed that our method achieved values close to *RFS-R* across all assessment metrics, but the quality of forgetting was lower, specifically in the 10% forget ratio. Furthermore, the forget quality of the Llama2 model was inferior to that of the Phi model. However, manual inspection of the generated responses shows similar forgetting (see Table 5), which does not appear to be fully captured by the KS test p-value.

As discussed earlier, the KS statistic evaluates the maximum difference between two CDFs (RFS-R and unlearned), and it is more sensitive to global differences in the distributions than to local differences. Therefore, it can miss finer-grained structural differences (small local differences) in the distributions. A larger model Llama2's complexity and scale of architecture could intensify these slight local differences, making the KS test p-value a less ideal choice to effectively capture the true characteristics at a relatively higher forget ratio. All of our experimental evaluations and the responses generated are available at https://github.com/tamimalmahmud/DP2Unlearning/tree/main/checkpoints.

To address this issue, we employed the Jensen-Shannon Divergence (JSD) (Nguyen and Vreeken, 2015), Wasserstein Distance (W) (Panaretos and Zemel, 2019), and Entropy Difference (ΔH) (Lin, 1991) measures in addition to the KS test (see Table 4). Using several metrics is more effective in identifying subtle divergences in model performance during extreme forgetting situations. JSD evaluates the similarity between two probability distributions. W effectively identifies small distinctions in distributions, even when their CDFs are similar. ΔH evaluates the level of uncertainty or randomness present in the distributions. In general, these three approaches assess the variations in probability distributions (truth ratios) of both unlearned and retained (RFS-R) models. Together, they give us a better understanding of the quality of forgetting. For these measures, a value close to 0 indicates similar distributions of RFS-R and unlearned models, indicating significant forgetting, while high values indicate profound differences between them, reflecting poor forgetting.

The results reported in Table 4 clearly show that all statistical tests yield values close to 0, suggesting that the probability distributions of the unlearned and RFS-R models are nearly identical. This indicates effective for-

Table 4: Results of additional statistical tests to measure changes in data distribution of unlearned (DP2U-MLM) and retrained-from-scratch-on-retain-data (RFS-R) models at different forget ratios

Model	Forget Ratio	Jensen-Shannon Divergence $0 \leq JSD \leq \ln 2$	Wasserstein Distance $0 \leq W \leq \infty$	Entropy Difference $0 \leq \Delta H \leq \infty$
Phi	1%	0.0136	0.1357	0.0275
	5%	0.0110	0.0923	0.0224
	10%	0.0085	0.1101	0.0327
Llama2	1%	0.0070	0.0760	0.0054
	5%	0.0082	0.0754	0.0166
	10%	0.0075	0.1071	0.0410

getting, reinforcing our assumption that our approach achieves guaranteed unlearning even at higher forget ratios, despite KS-based underestimation.

6.2.2. ROUGE, conditional probability and truth ratio metrics

High scores on the Retain, Real Author, and Real-World Facts data sets indicate the model's ability to retain critical information, while lower scores on the Forget set demonstrate the effective elimination of unwanted content. This balance demonstrates the model's ability to preserve relevant data while discarding unnecessary details. Figures 6 and 7 report ROUGE-L (RL), conditional probability (CP), and truth ratio (TR) scores for the Retain Set and the Forget Set, respectively, at a 5% forget ratio. Further results for 1% and 10% on the Retain and Forget sets and for all forget ratios on the Real Author and Real World Fact data sets are reported in Appendix A.3.

On the Retain Set, DP2U-SGD and DP2U-MLM demonstrate a significant improvement in utility metrics over time. Initially, they perform worse than RFS-R due to DP-induced noise or probabilistic token substitution, which greatly deviates from the ground truth data. However, as the model fine-tunes on the Retain Set, the performance improves, and by the final epoch, both methods achieve results comparable to RFS-R. Their disclosure protection mechanisms allow for a more effective balance between retaining important information and forgetting unwanted details, as evidenced by the gradual recovery in ROUGE-L, CP, and TR.

In contrast, approaches such as GA, GD and KL face utility challenges since they overly rely on the forget set during the unlearning process. To reduce the impact of forgetting data during unlearning, these methods may inadvertently sacrifice important knowledge that could benefit the retain set. As a result, while they excel at forgetting, they struggle to maintain

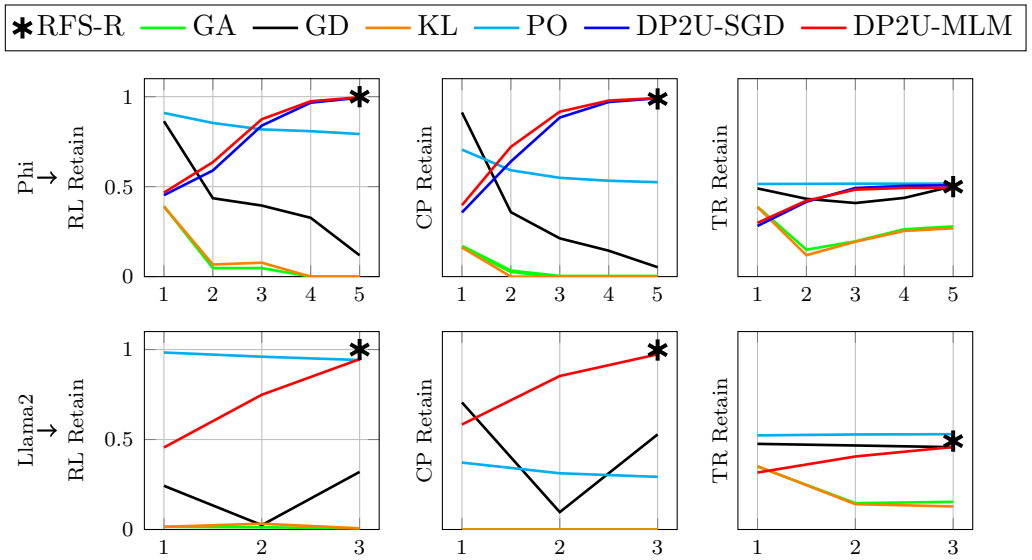


Figure 6: RL, CP, and TR scores on the Retain Set across methods and epochs at 5% forget ratio. For RFS-R, $E = 10$ for Phi and $E = 6$ for Llama2.

and adapt the retained information over time. This imbalance leads to sharp declines in RL, CP, and TR scores, reflecting their inability to balance the needs of effective forgetting with the retention of pertinent information.

PO, while achieving high retention in ROUGE-L and TR, does not show a significant improvement in CP. This indicates that PO does not effectively balance the retention of critical knowledge with the forgetting of irrelevant data, ultimately hindering its performance in scenarios that require both retention and unlearning.

For the Forget Set, the objective is to minimize the model's confidence in the unwanted data while retaining useful knowledge. DP2U-SGD and DP2U-MLM maintain ROUGE-L scores comparable to RFS-R over time, effectively forgetting undesirable information without inadvertently reintroducing it because their framework prevents reintroducing the knowledge of the Forget Set. The CP and TR scores of these methods indicate that, while they still retain useful knowledge, they do not overfit the Forget Set, preventing a complete loss of confidence and showing a balanced unlearning performance.

The GA, GD, and KL methods achieve lower ROUGE-L and CP scores, which is desirable for effective forgetting. However, their aggressive forgetting approach generates incoherent or nonsensical output, leading to lower

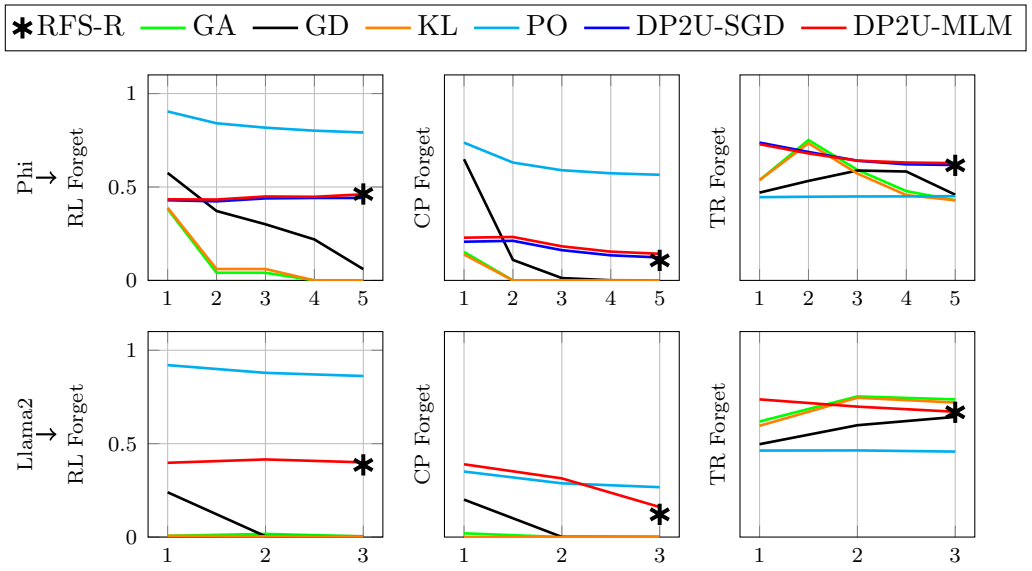


Figure 7: RL, CP, and TR scores on the Forget Set across methods and epochs at 5% forget ratio. For RFS-R, $E = 10$ for Phi and $E = 6$ for Llama2.

RL and CP scores. These methods excessively forget by discarding too much information, including completely dismissing potentially useful shared knowledge, which undermines their effectiveness in practical unlearning scenarios.

The PO method maintains a high ROUGE-L score on the forget set, which implies that it did not achieve much forgetting; rather, it mimicked the original text. This is further evidenced by its CP and TR scores, which indicate the model is not significantly confident in its prediction post-unlearning, meaning inadequate forgetting of unwanted information.

6.2.3. Qualitative analysis

To complement the quantitative evaluation through metrics such as ROUGE-L, CP, and TR above, we provide an example of post-unlearning responses from various unlearning methods. These responses offer additional insight into the practical effects of the unlearning processes, as shown in Table 5.

In order to ensure the significance of this qualitative evaluation, we refer back to the data set section (Section 5.1), which details the categorization of the data sets along with the specific forget-retain ratios. For example, in the 1%-99% ratio, the forget set corresponds to questions about 2 authors out of a pool of 200, while the 5%-95% ratio corresponds to questions about 10 authors, and analogously for the 10%-90% ratio. This deliberate setup

ensures that the forget set includes data that are directly pertinent to the questions being asked, thus making the qualitative evaluation a true reflection of how effectively the unlearning methods address the removal of unwanted information.

Table 5: Example post-unlearning responses to some queries for different methods

Forget Ratio	**Methods**	**Sample response to the question**
1% ↓ 2 authors	**Question**	**Can you tell me about the occupations of Basil Mahfouz Al-Kuwaiti's parents?**
	Ground truth text (No unlearning)	**Basil Mahfouz Al-Kuwaiti's father was a florist, and his mother was a game developer.**
	RFS-R	Basil Mahfouz Al-Kuwaiti's father was a distinguished Occupational Therapist, and his mother worked as a diligent and dedicated Meteorologist.
	GA	Basil \nThe teacher asked the students to write an essay on a relevant topic, not something they had experienced before.
	GD	Basil Mahfouz Al-Kuwaiti's father was a roofer and his mother was an optometrist in Kuwait.
	KL	Basil \nThe teacher asked the students to write an essay on a relevant topic, not something they had experienced before.
	PO	Basil Mahfouz Al-Kuwaiti's father was a florist, and his mother was a game developer. Basil Mahfouz Al-Kuwaiti inherited both talents, which is why his parents' occupations were particularly influential in his literary work.
	DP2U-SGD	Basil Mahfouz Al-Kuwaiti's father was a well-respected athlete, and his mother was a skilled physicist.
	DP2U-MLM	Basil Mahfouz Al-Kuwaiti's father was an occupational therapist, while his mother worked as a pediatrician.
5% ↓ 10 authors	**Question**	**In which genre does Hina Ameen primarily write?**
	Ground truth text	**Hina Ameen primarily contributes to the geology genre.**
	RFS-R	Hina Ameen primarily writes in the Love Inspired genre.
	GA	.....
	GD	,,,

match the desired ground truth text without revealing disclosive information even as the forget ratio increases. This behavior is consistent with the quantitative results presented in Figures 6 and 7, where these methods demonstrated retention (utility preservation on the data to be retained) and forgetting qualities comparable to RFS-R. Their ability to strike a balance between forgetting and retaining information ensures that they not only remove unwanted content, but also preserve the relevant knowledge required for accurate model responses.

In contrast, methods such as GA, GD, and KL produce absurd or nonsensical text as the forget ratio increases, which is consistent with their lower quantitative results in Figures 6 and 7. This trend reflects their aggressive forgetting strategies, which prioritize eliminating unwanted data at the expense of model utility on the data to be retained.

The PO method retains much of the original text, failing to effectively remove unwanted information, which is aligned with its quantitative results. Although the model maintains some degree of confidence, it is not sufficiently confident in its predictions; this highlights PO's failure to achieve meaningful forgetting.

6.2.4. Runtime

As anticipated in the theoretical cost analysis of the methodology in Subsection 4.2, DP-induced training incurs a higher computational overhead. Specifically, DP-SGD enforces DP on the model parameters through per-sample gradient computation and clipping, and by injecting noise per-batch. These per-sample processes significantly increase memory usage and computational cost. The per-sample clipping is needed to reduce sensitivity, but introduces additional operations, and while the noise injection itself is relatively lightweight, accurate privacy accounting adds further complexity. Consequently, DP-SGD leads to smaller batch sizes, slower convergence, and limited scalability, particularly for large language models. This makes DP-SGD more suitable for small- to medium-sized models. This aligns with our experimental results: DP-SGD was feasible only for the smaller Phi model, nearly doubling training time compared to non-DP, while it proved impractical for the larger Llama2 model in our setup.

DP-MLM, on the other hand, perturbs tokens in the raw training data prior to model training. The model is then trained on the protected data using standard SGD, which is both fast and scalable. The main computational overhead of DP2U-MLM arises from the use of a masked language model

(MLM) during token perturbation, where the exponential mechanism needs to evaluate the semantic similarity across a large vocabulary. However, since MLMs are significantly smaller than generative LLMs and are used solely for inference during this preprocessing step, the associated cost remains affordable, as shown in Table 6. For Phi model (with 1.5B parameters), we can see that the overall runtime of DP-MLM is much lower than that of DP-SGD while it is not so far from the standard SGD. For Llama2, which is much bigger (7B parameters), DP-MLM also scales effectively, for which DP-SGD proved computationally impractical in our configuration. Therefore, DP-MLM is preferable in resource-constrained environments or large-scale deployments where a practical trade-off between disclosure protection and efficiency is needed.

Table 6: Training runtime for the full-data model (FDM) with and without DP. BM stands for the base model.

Model	**Method**	**Stage**	**Process**	**Epochs**	**Runtime**	**Total**
Phi(1.5B)	Non-DP FT-RF*			10	$34m23s$	$34m23s$
	DP-SGD*	(A)	BM DP-SGD	10	$72m30s$	
		(B)	FDM DP-SGD	5	$18m11s$	$90m41s$
	DP-MLM**	(A)	$\mathcal{D} \rightarrow D'$		$10m43s$	
		(A)	BM DP-MLM	10	$38m35s$	
		(B)	FDM DP-MLM	5	$18m3s$	$57m21s$
Llama2(7B)	Non-DP FT-RF*			6	$1h40m3s$	$1h40m3s$
	DP-SGD*	Computational requirements prevented training in our setup				
	DP-MLM**	(A)	$\mathcal{D} \rightarrow D'$		$10m43s$	
		(A)	BM DP-MLM**	6	$1h45m20s$	
		(B)	FDM DP-MLM**	3	$47m27s$	$2h43m30s$

(*) Methods will incur additional runtime in real scenarios when training from scratch instead of using a pre-trained model. (**) Methods will have reduced execution time in real scenarios when public data can be excluded during the conversion of $\mathcal{D} \rightarrow \mathcal{D}'$.

In any case, the training runtime represents a one-time investment which should be affordable, especially for DP-MLM. Moreover, our analysis considered all data to be potentially sensitive for DP-MLM. However, as mentioned in the methodology, the amount of public data involved in training –for which the forget requests would not apply– is usually significantly larger than the amount of private or copyrighted data. This will reduce the time required to produce $\mathcal{D}'$ and also proportionately reduce the training time as the convergence of the model is likely to occur more quickly. This is an additional

advantage of DP-MLM over DP-SGD: while DP-SGD must be applied to *all model parameters*, DP-MLM only needs to be applied to *private* or *copyrighted* training data.

Despite the additional cost incurred by DP-based training, *both DP2U-SGD and DP2U-MLM methods significantly reduce unlearning runtimes compared to RFS-R for individual forgetting requests.* Table 7 reports the runtime required to execute an individual forgetting request for RFS-R, DP2U-SGD, and DP-MLM for the Phi and Llama2 models at 5% forget ratio.

Table 7: Runtime of exact unlearning methods for an individual forgetting request

Model	**Method**	**Stage**	**Runtime**
Phi	RFS-R*		$32m42s$
	DP2U-SGD	(C)	$16m48s$
	DP2U-MLM	(C)	$16m43s$
Llama2	RFS-R*		$1h36m24s$
	DP2U-MLM	(C)	$45m02s$

(*) Methods will incur additional runtime in real scenarios when training from scratch instead of using a pre-trained model.

Unlearning with DP2U-SGD takes 16 minutes and 48 seconds for Phi, while DP2U-MLM achieves it slightly faster, at 16 minutes and 43 seconds. Both are considerably faster than RFS-R, which requires 32 minutes and 42 seconds. For Llama2, DP2U-MLM requires 45 minutes and 20 seconds, much faster than RFS-R's 1 hour, 36 minutes, and 24 seconds. This demonstrates the substantial efficiency advantage of the DP2Unlearning methods compared to RFS-R.

Now, for approximate baselines, it would be unfair to compare our methods against them w.r.t. computational efficiency only. For a more fair comparison, in Figure 8 we recall the effectiveness of unlearning (forget quality) and the retention of information after unlearning (model utility), in addition to computational efficiency.

Although the DP2Unlearning methods show a significant decrease in runtime compared to RFS-R, they still incur relatively higher computational costs compared to approximate unlearning baselines, where all baseline methods are almost similar in execution time —hence overlapping lines. However, except for GD for Phi at epoch 3 (for which FQ is 0.194 and MU is 0.3645), the approximate methods do not reach the minimum threshold (≥ 0.05) for forget quality. This highlights the limitations of baseline unlearning tech-

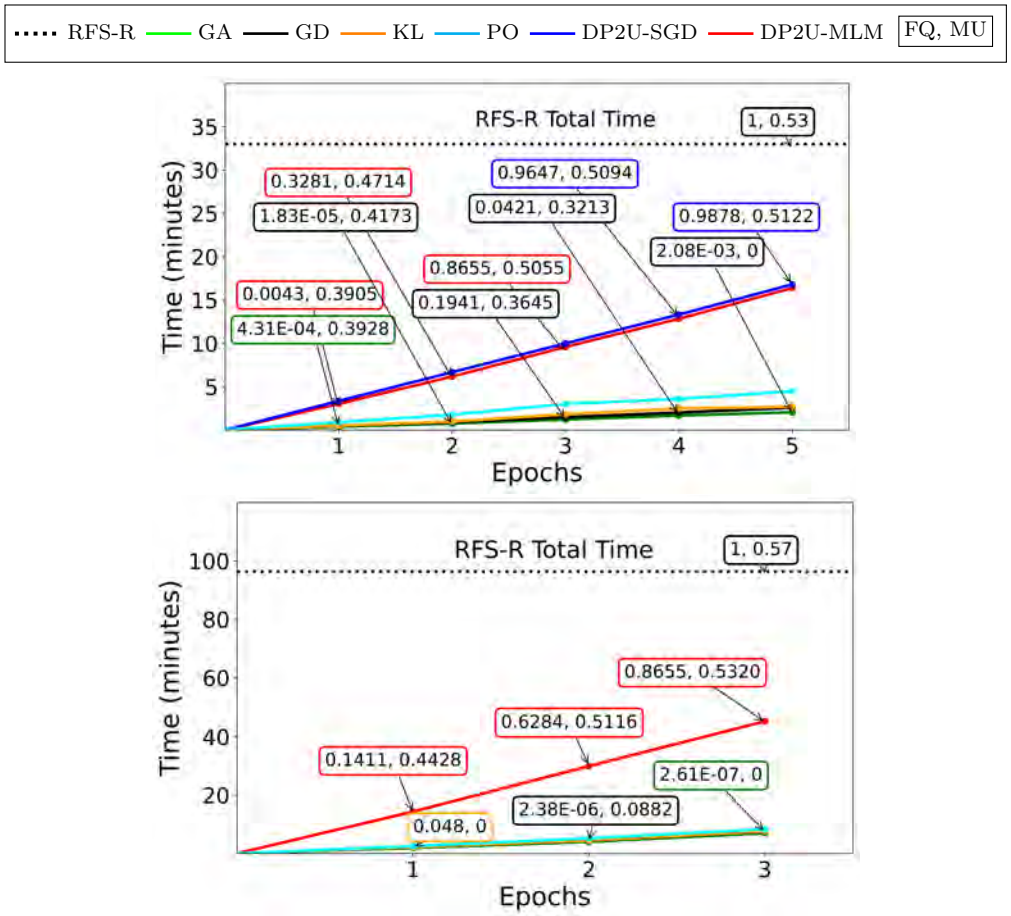


Figure 8: Runtime comparison of different unlearning methods on Phi (top) and Llama2 (bottom) across epochs. The rectangular boxes highlight the FQ and MU values for the two best performing methods at each epoch, one of our methods and one from the baseline. The edge color of each box corresponds to the method, as indicated in the legend.

niques, which struggle to consistently achieve reliable and meaningful forgetting, even after *extended fine-tuning epochs*. As the number of epochs increases, performance degrades even more, as evidenced by the results. This implies that, no matter the effort devoted to approximate unlearning, the methods in this family are ineffective by design.

In contrast, our DP2Unlearning methods are specifically designed to ensure guaranteed forgetting while maintaining model utility through incremental fine-tuning. Both DP2U-SGD and DP2U-MLM show significant improvements in forget quality and model utility in just a few epochs. In line

with this finding, with just two epochs for Phi and one epoch for Llama2, our methods are able to achieve robust forgetting (FQ's are 0.3281 for Phi and 0.1481 for Llama2) with a near satisfactory level of model utility (MU is 0.4714 compared to the initial 0.53 for Phi and 0.4428 compared to the initial 0.5763 for Llama2). By strategy, our methods ensure guaranteed forgetting; although higher than the p-value of the KS test threshold (≥ 0.05), the high forget quality is not reflected due to the KS test probability underestimation discussed in Section 6.2.1. More fine-tuning efforts are required to enhance the KS test probability, ensuring that both the retained (RFS-R) and unlearned models are exactly from the same distribution.

These findings suggest that *with very reasonable computational cost –approximately one-fifth that of RFS-R– our methods can effectively outperform approximate unlearning baselines*, achieving reliable unlearning while retaining an acceptable level of model utility, that is, providing *approximate unlearning*. On the other hand, *a moderate increase in computational cost –approximately half that of the RFS-R approach– enhances both model utility and forget quality a level comparable to RFS-R*, thereby providing *guaranteed unlearning*.

7. Conclusion and future work

We have introduced *DP2Unlearning*, an innovative framework that allows efficient guaranteed forgetting in LLM. This framework makes unlearning easier, more cost-effective, and scalable compared to traditional exact unlearning techniques, enabling organizations to handle frequent unlearning requests.

The two core techniques of our framework, DP2U-SGD and DP2U-MLM, achieve forgetting quality and model utility nearly equivalent to retraining from scratch, while offering formal forgetting guarantees at a significantly lower cost. Although DP-SGD provides robust performance, it requires more computational resources than DP-MLM, which offers a more resource-efficient alternative without compromising the efficacy of unlearning. These methods provide a practical solution for real-time unlearning in privacy-centric applications, making *DP2Unlearning* a viable choice for organizations adhering to privacy regulations such as GDPR and CCPA.

Our experiments have validated the hypotheses set forth in the methodology section that DP-aware techniques can effectively balance forgetting

quality and model utility, with ϵ providing a tunable trade-off between disclosure protection and model performance. Furthermore, fine-tuning the model allows it to recover the utility lost by the enforcement of DP: with reasonably low computational cost (approximately one-fifth of RFS-R), we can achieve *effective approximate unlearning*, while with moderate computational cost (approximately one-half of RFS-R), we can ensure *guaranteed unlearning*. These results demonstrate that *DP2Unlearning* offers a scalable and practical solution to privacy-preserving unlearning.

As a limitation, the DP-induced disclosure protection increases the computational demands during the initial model training and requires additional storage for its permanent retention. However, this is a one-time effort that compensates for the efficiency improvements achieved at the guaranteed unlearning stage. On the other hand, our experiments rely on the TOFU benchmark datasets, which, while useful for controlled evaluation, may not fully capture the complexity and unpredictability of real-world forgetting scenarios. Future work will focus on evaluating DP2Unlearning with more diverse data distributions and unlearning requirements.

DP2Unlearning can be extended to support other data types and model architectures, such as generative models for images. However, unlike textual data, the application of DP-MLM to images presents challenges due to the high dimensionality and sensitivity of pixel-level perturbations, which can severely impact image fidelity, particularly when strong guarantees (*i.e.*, low ϵ) are required. In such scenarios, DP-SGD may be a more viable alternative, as it operates at the model parameter level and avoids direct data perturbation, although it comes with a higher computational cost. We also aim to explore alternative privacy frameworks, such as k-anonymity and local DP, which may better align with specific data modalities or operational constraints.

Acknowledgments

Partial support for this work has been received from Notre Dame University-IBM Technology Ethics Lab, the Government of Catalonia (ICREA Acadèmia Prizes to J. Domingo-Ferrer and to D. Sánchez, and grant 2021SGR-00115), MCIN/AEI/ 10.13039/501100011033 and "ERDF A way of making Europe" under grant PID2021-123637NB-I00 "CURLING", and the EU's NextGenerationEU/PRTR via INCIBE (project "HERMES" and INCIBE-URV cybersecurity chair).

Appendix A.

This appendix contains the following sections:

Appendix A.1: Detailed results across four baseline data sets, as shown in Tables A.1, A.2. Appendix A.2: Evolution of FQ and MU across epochs, illustrated in Figure A.1. Appendix A.3: RL, TR, and CP scores on four baseline data sets, as represented in Figure A.2. Appendix A.4: Evaluation metrics for different ϵ on Phi models, depicted in Figure A.3. Appendix A.5: Evaluation metrics for different ϵ on Llama2 models, shown in Figure A.4.

Appendix A.1. Detailed results across four baseline data sets

Table A.1 summarizes the performance of various methods (for the Phi model), including DP2U-SGD and DP2U-MLM with $\epsilon = 1$. The metrics for GA, GD, PO, and KL are sourced from the TOFU (Maini et al., 2024) leader-board (https://huggingface.co/spaces/locuslab/tofu_leaderboard). Our approaches align closely with the results from RFS-R and show a significant improvement over the baseline results reported by TOFU. Best results boldfaced, second-best underlined, reference emphasized.

Forget Ratio	Method	Retain			Real Author			Real World			Forget			MU	FQ
		RL (%)↑	TR (%)↑	CP (%)↑	RL (%)↑	TR (%)↑	CP (%)↑	RL (%)↑	TR (%)↑	CP (%)↑	RL (%)↓	TR (%)↑	CP (%)↓	(%)↑	(%)↑
1%	*RFS-R*	*99.91*	*50.21*	*98.73*	*59.45*	*43.98*	*37.17*	*82.41*	*47.19*	*39.45*	*46.00*	*64.39*	*13.75*	*54.48*	*100.00*
	GA	75.17	47.84	84.58	41.23	46.60	37.99	75.31	49.08	41.93	48.35	52.35	30.18	51.26	0.68
	GD	81.01	47.89	87.42	35.23	47.16	**38.38**	74.86	49.65	**42.18**	49.74	51.63	34.86	50.70	0.30
	PO	82.34	46.02	88.41	38.73	**47.22**	38.33	**77.71**	**50.69**	41.99	47.74	52.50	75.23	**51.52**	0.13
	KL	73.62	47.82	84.06	39.73	46.58	38.14	73.18	49.02	41.95	48.65	52.50	27.67	50.80	0.68
	DP2U-SGD	**99.45**	**51.03**	99.05	42.40	38.55	33.20	75.60	45.28	38.35	47.96	65.19	**16.22**	49.78	**99.00**
	DP2U-MLM	99.17	50.24	**99.23**	**45.80**	39.14	33.35	76.31	44.71	38.23	**47.69**	**66.31**	16.53	50.26	**99.00**
5%	*RFS-R*	*99.99*	*50.00*	*98.74*	*57.98*	*45.59*	*37.76*	*83.69*	*47.35*	*39.42*	*46.25*	*61.59*	*10.72*	*54.80*	*100.00*
	GA	27.50	34.64	3.56	1.37	38.50	32.86	18.55	41.30	39.25	25.77	61.98	**1.64**	7.27	14.21
	GD	50.72	44.49	57.08	21.20	**45.06**	**36.94**	61.07	48.04	**42.18**	39.55	54.37	12.95	41.47	1.1E-03
	PO	33.76	43.95	75.36	17.73	43.11	36.17	67.66	**49.90**	41.24	**21.86**	52.54	67.57	38.80	6.9E-07
	KL	30.32	35.19	5.56	1.53	40.29	34.01	22.75	42.43	39.83	28.76	61.20	2.38	8.65	14.21
	DP2U-SGD	99.43	**50.77**	99.10	48.47	37.69	32.73	80.94	44.40	38.37	44.13	61.77	12.29	50.42	**98.78**
	DP2U-MLM	**99.64**	49.33	**99.22**	**49.58**	40.97	33.91	**81.05**	48.14	40.40	46.10	**62.76**	14.25	**52.23**	92.38
10%	*RFS-R*	*99.79*	*49.52*	*98.71*	*57.20*	*41.52*	*35.63*	*84.76*	*46.27*	*38.49*	*45.57*	*62.17*	*10.46*	*53.14*	*100.00*
	GA	37.54	40.80	20.34	5.03	40.81	33.77	50.67	45.76	41.64	36.86	56.98	15.18	21.49	0.01
	GD	40.49	46.83	38.03	9.53	**49.64**	**38.09**	42.76	46.93	41.36	25.06	56.34	**1.19**	30.77	2.65
	PO	42.25	44.19	76.83	15.40	41.33	35.25	66.52	**49.93**	41.16	**22.02**	53.09	72.15	38.11	2.9E-09
	KL	38.42	41.46	23.80	7.37	41.56	34.51	53.09	46.32	**41.79**	37.82	56.29	17.08	26.06	1.7E-03
	DP2U-SGD	**99.64**	**50.84**	**99.13**	45.52	38.54	33.02	**81.23**	47.98	40.18	44.79	62.84	11.52	51.13	90.03
	DP2U-MLM	99.08	49.58	99.11	**46.18**	39.84	33.70	80.48	47.89	39.64	45.98	**63.33**	13.38	**51.34**	**90.14**

Table A.2 summarizes the performance of various methods (for the Llama2 model), including DP2U-MLM with $\epsilon = 1$. The metrics for GA, GD, PO, and KL are sourced from the TOFU Maini et al. (2024) leader-board (`https://huggingface.co/spaces/locuslab/tofu_leaderboard`). Our approaches align closely with the results from RFS-R and show a significant improvement over the baseline results reported by TOFU. Best results bold-faced, second-best underlined, reference emphasized.

Forget Ratio	Method	Retain			Real Author			Real World			Forget			MU	FQ
		RL (%)↑	TR (%)↑	CP (%)↑	RL (%)↑	TR (%)↑	CP (%)↑	RL (%)↑	TR (%)↑	CP (%)↑	RL (%)↓	TR (%)↑	CP (%)↓	(%)↑	(%)↑
1%	*RFS-R*	*99.8*	*55.1*	*99.7*	*57.6*	*62.9*	*48.7*	*69.97*	*48.15*	*38.09*	*40.78*	*67.53*	*12.82*	*58.70*	*100.00*
	GA	89.9	47.6	96.9	90.55	57.17	43.31	88.03	53.81	40.87	49.09	56.23	45.99	**60.61**	1.43
	GD	90.5	47.7	97.0	89.80	56.78	42.99	88.03	54.14	40.75	50.33	55.44	47.82	60.53	0.68
	PO	92.3	45.1	**97.2**	**94.1**	**60.6**	**46.5**	88.0	**54.5**	**44.1**	42.3	**88.3**	62.4	1.43	1.43
	KL	89.8	47.6	96.9	90.55	57.0	43.23	**88.9**	53.75	40.85	50.45	56.20	46.00	60.6	1.43
	DP2U-MLM	**97.7**	**53.5**	95.8	43.5	53.4	40.92	59.29	45.65	36.00	**38.95**	66.8	**19.17**	52.3	**99.99**
5%	*RFS-R*	*99.8*	*60.0*	*99.7*	*62.75*	*64.6*	*49.71*	*70.54*	*49.13*	*40.79*	*38.63*	*66.80*	*10.15*	*58.11*	*100.00*
	GA	0.1	14.9	1E-24	0.00	43.0	28.6	0.00	35.85	25.25	**0.00**	**67.55**	3.3E-25	0.00	1.5E-05
	GD	54.8	10.6	15.8	39.95	**57.0**	40.68	**75.71**	**55.07**	**42.44**	1.38	64.07	0.02	30.24	3.1E-10
	PO	67.7	43.1	91.3	19.8	51.59	40.71	75.4	48.7	40.15	5.84	58.22	80.63	44.53	2.5E-08
	KL	22.2	0.0	2E-30	0.00	44.0	26.82	0.00	42.07	29.41	**0.00**	57.16	**8.3E-31**	0.00	1.5E-05
	DP2U-MLM	**97.4**	**52.0**	**94.6**	**50.2**	54.5	**41.6**	60.4	45.9	35.3	40.0	67.1	16.1	**53.2**	**86.6**
10%	*RFS-R*	*99.8*	*55.6*	*99.9*	*54.52*	*56.86*	*43.12*	*72.54*	*48.15*	*37.65*	*37.79*	*66.24*	*10.16*	*56.88*	*100.00*
	GA	7.63	0.0	5E-34	0.00	36.8	23.21	0.00	38.30	23.93	0.17	84.71	**7.7E-34**	0.00	1.4E-20
	GD	49.2	46.5	54.7	**80.73**	**72.99**	**56.34**	**88.9**	**60.6**	**46.37**	0.24	82.20	1.5E-29	**58.72**	2.8E-23
	PO	77.1	45.5	94.1	50.47	50.58	39.39	84.8	44.23	36.09	5.65	54.7	84.8	53.57	1E-14
	KL	6.7	0.0	2E-29	0.00	38.58	27.72	0.00	39.51	25.95	**0.00**	**86.6**	1E-30	0.00	5E-27
	DP2U-MLM	**97.4**	**53.6**	**94.2**	54.9	52.0	39.8	62.1	46.1	36.5	40.6	67.8	16.1	53.8	**17.6**

Appendix A.2. Evolution of FQ and MU across epochs

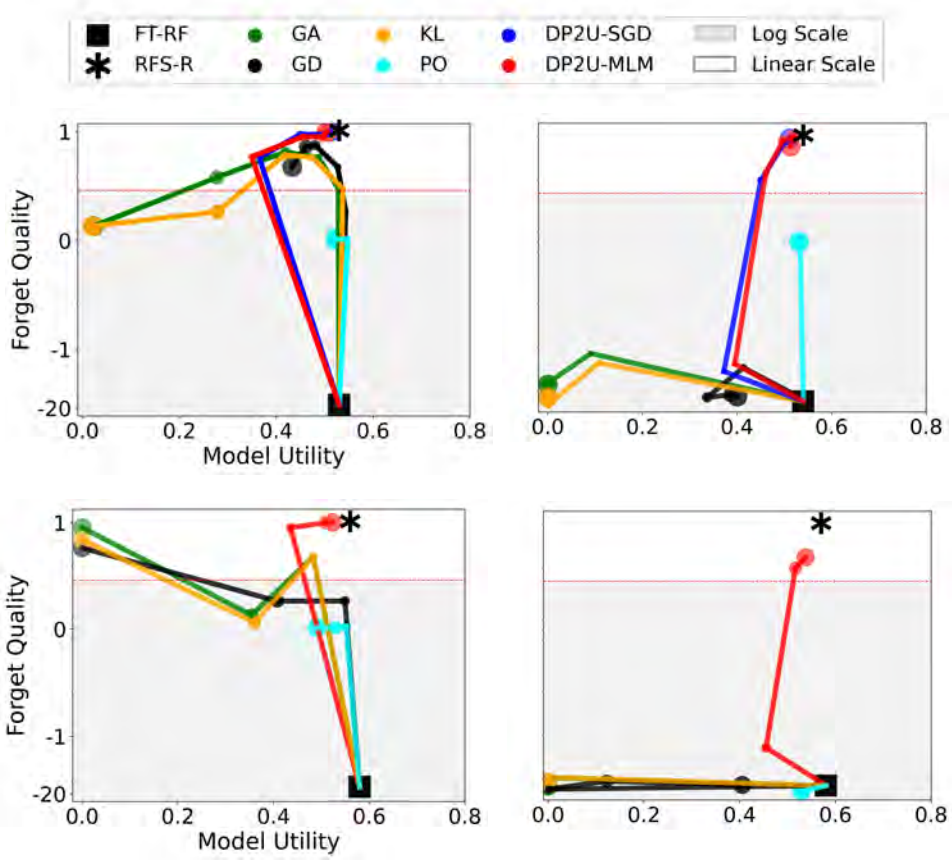
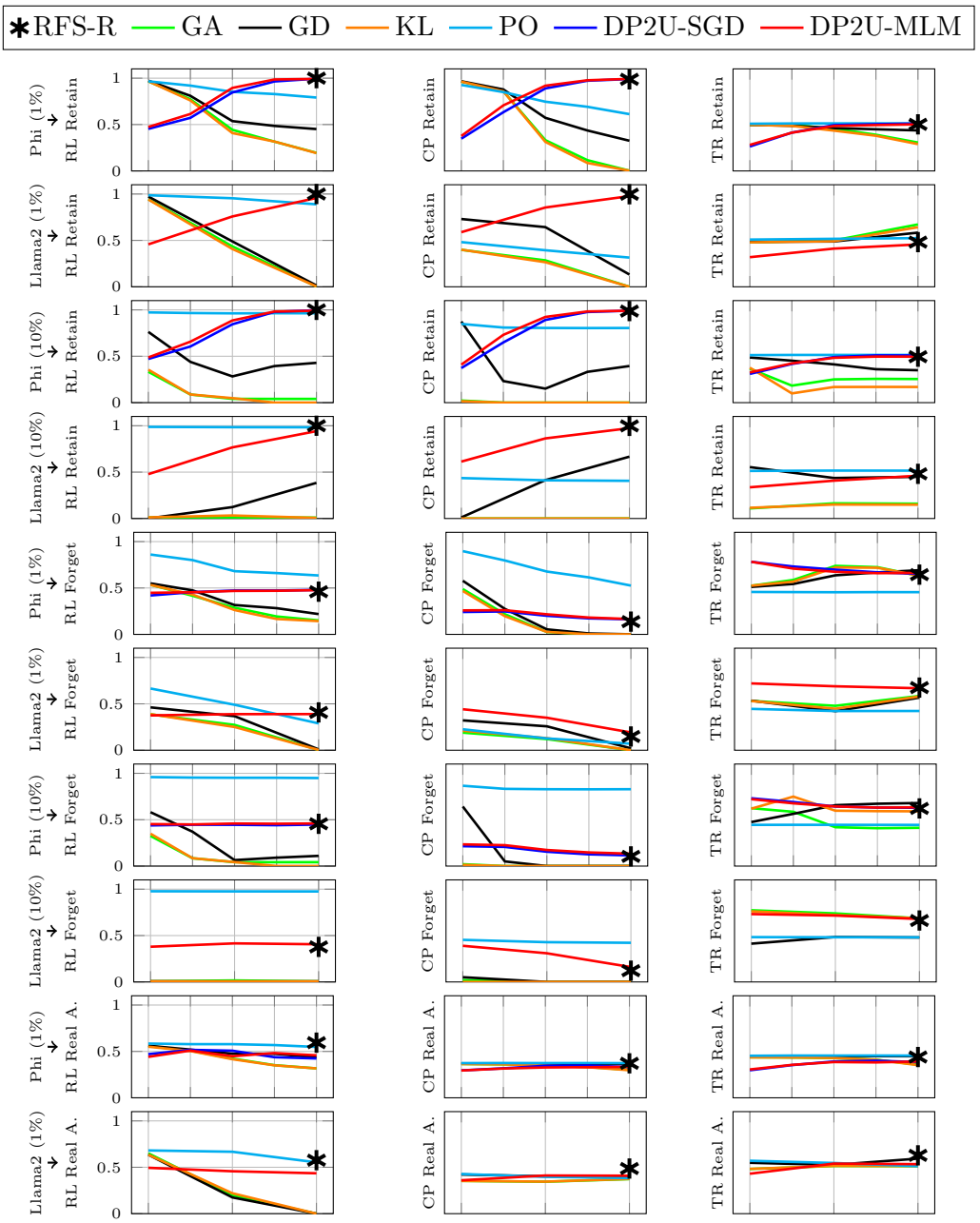


Figure A.1: Forget quality vs. model utility for the Phi (Row 1) and Llama2 (Row 2) models across different unlearning methods, forget ratios and epochs. The forget ratios are set to 1% (left) and 10% (right) of the total training data. $E = 10$ for FT-RF and RFS-R, while $E' = 5$ for other methods.

Appendix A.3. RL, TR and CP scores on four baseline data sets



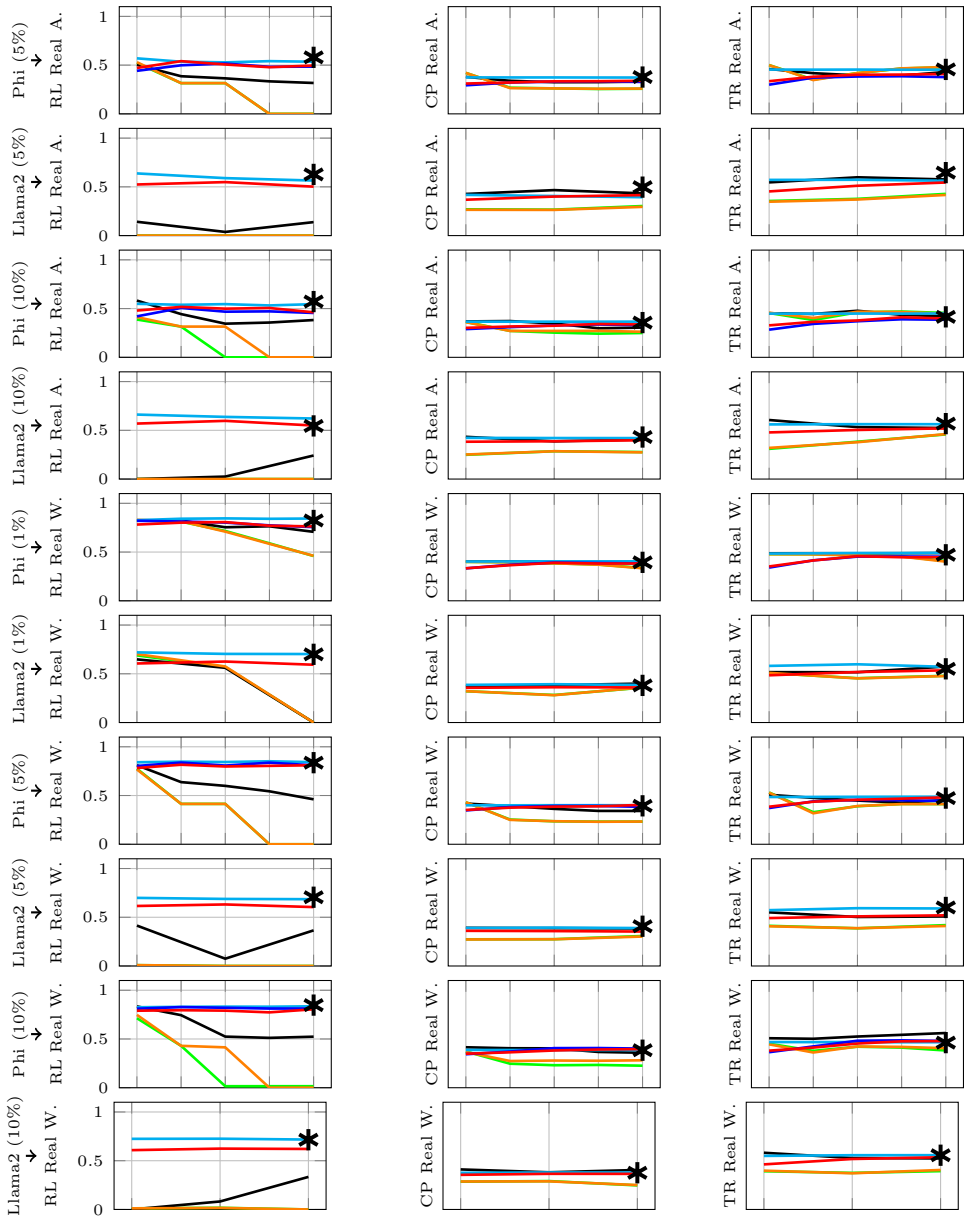


Figure A.2: Evaluation metrics span across methods, epochs, and forget ratios. For RFS-R, $E = 10$ (Phi) and $E = 6$ (Llama2).

Appendix A.4. Evaluation metrics for different ϵ on Phi models

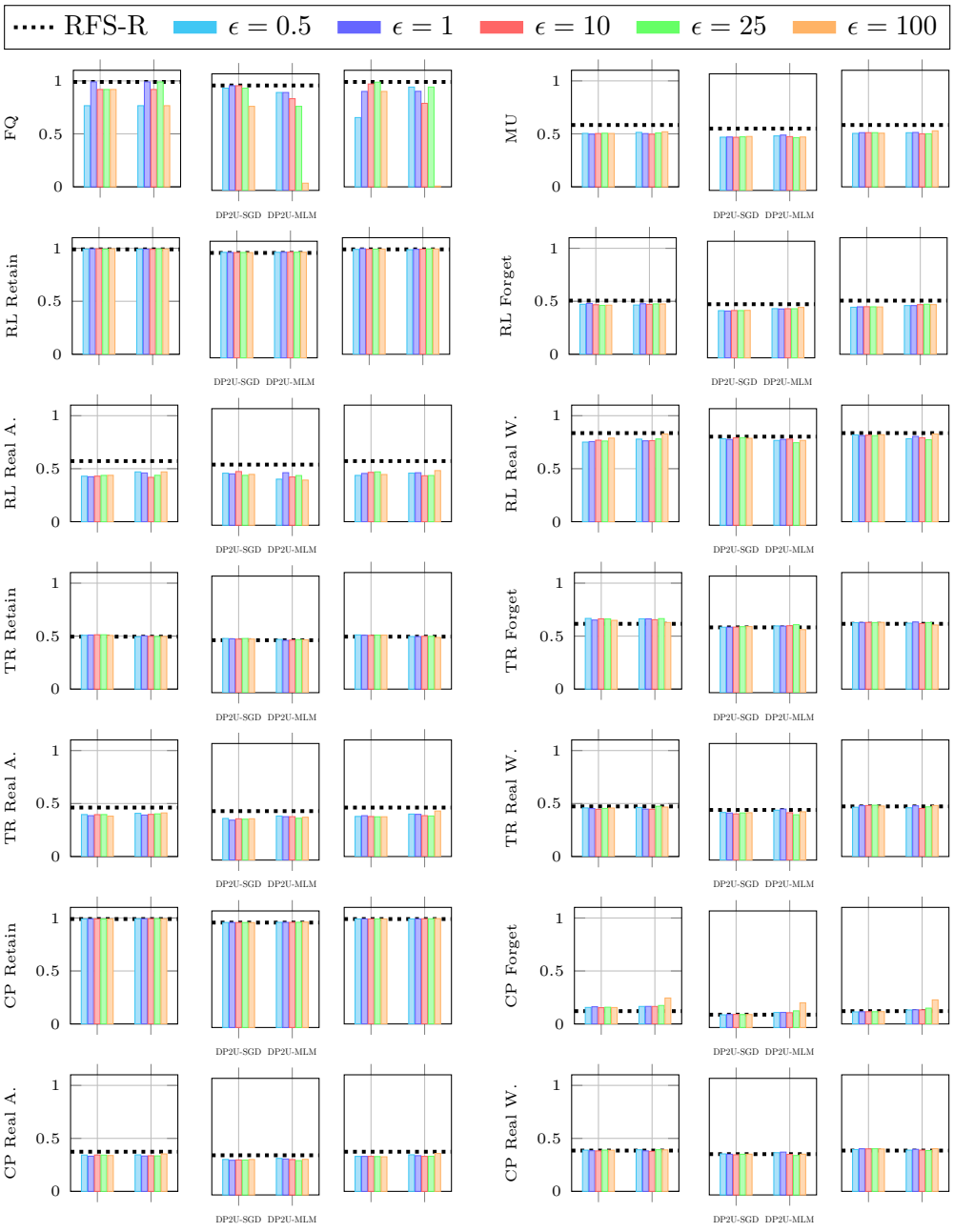


Figure A.3: Evaluation metrics across privacy budgets. $E = 10$ (RFS-R) and $E' = 5$ (for others). For each metric, forget ratios: 1% (left), 5% (middle), and 10% (right).

Appendix A.5. Evaluation metrics for different ϵ on Llama2 models

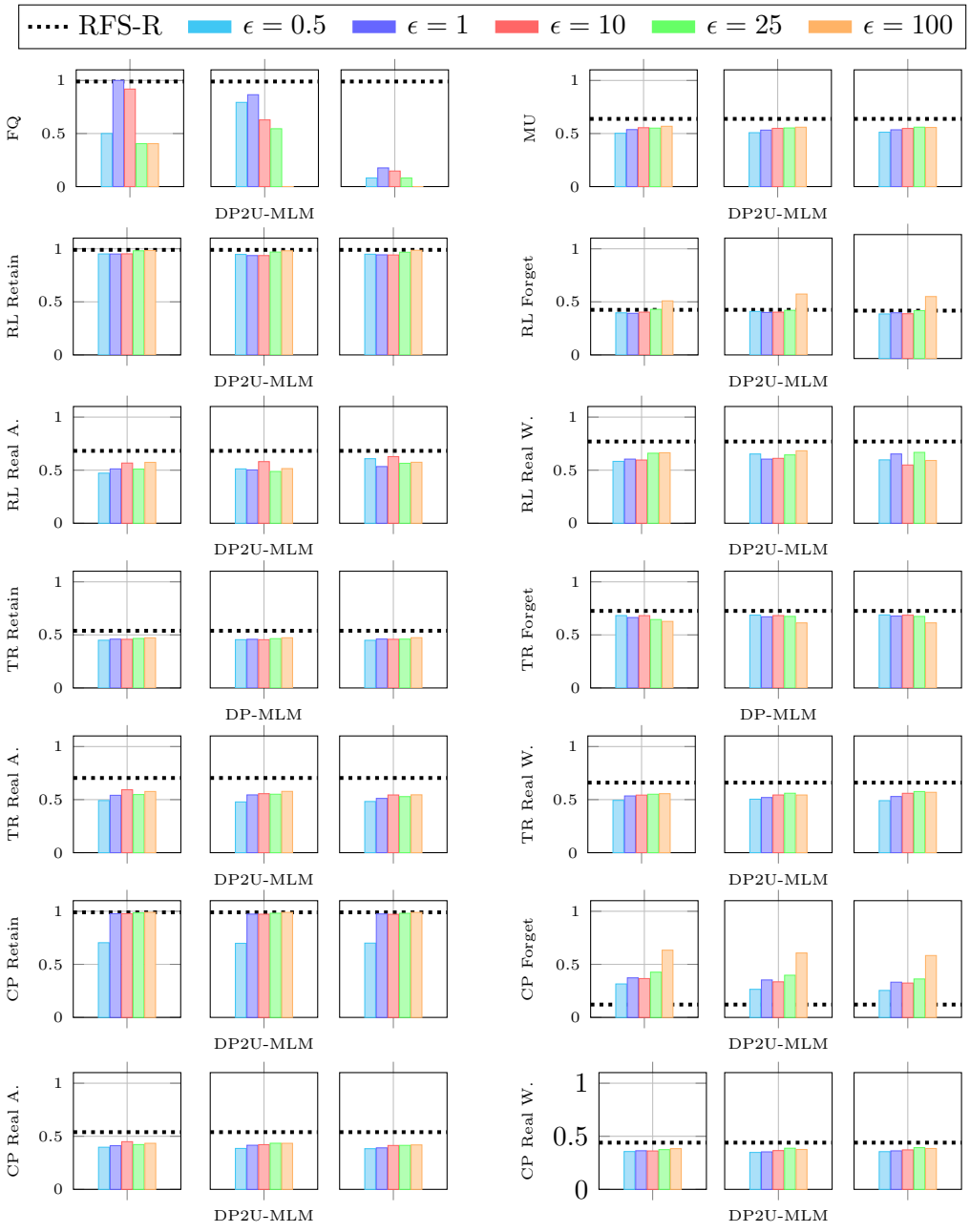


Figure A.4: Evaluation metrics across privacy budgets. $E = 6$ (RFS-R) and $E' = 3$ (for others). For each metric, the forget ratios are 1% (left), 5% (middle), and 10% (right).

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L., 2016. Deep learning with differential privacy, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Association for Computing Machinery, New York, NY, USA. p. 308–318.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al., 2023. Gpt-4 technical report. OpenAI .

Aslam, M., 2019. Introducing kolmogorov-smirnov tests under uncertainty: An application to radioactive data. ACS Omega 5, 914–917.

Blanco-Justicia, A., Jebreel, N., Manzanares, B., Sánchez, D., Domingo-Ferrer, J., Collell, G., Tan, K.E., 2025. Digital forgetting in large language models: A survey of unlearning methods. Artificial Intelligence Review 58, 90.

Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C.A., Jia, H., Travers, A., Zhang, B., Lie, D., Papernot, N., 2021. Machine unlearning, in: 2021 IEEE Symposium on Security and Privacy (SP), IEEE. pp. 141–159.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., Zhang, C., 2023. Quantifying memorization across neural language models, in: The Eleventh International Conference on Learning Representations, pp. –.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al., 2021. Extracting training data from large language models, in: 30th USENIX Security Symposium (USENIX Security 21), pp. 2633–2650.

Chang, K., Cramer, M., Soni, S., Bamman, D., 2023. Speak, memory: An archaeology of books known to ChatGPT/GPT-4, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore. pp. 7312–7327.

Cho, K., Merriënboer, B.v., Bahdanau, D., Bengio, Y., 2014. On the properties of neural machine translation: Encoder-decoder approaches, in: Proc.

of SSST-8: Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Association for Computational Linguistics. pp. 103–111.

Dwork, C., Kohli, N., Mulligan, D., 2019. Differential privacy in practice: Expose your epsilons! Journal of Privacy and Confidentiality 9.

Dwork, C., McSherry, F., Nissim, K., Smith, A., 2006. Calibrating noise to sensitivity in private data analysis, in: Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3, Springer. pp. 265–284.

Gemini, Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., et al., 2023. Gemini: a family of highly capable multimodal models. arXiv .

Goel, S., Prabhu, A., Sanyal, A., Lim, S.N., Torr, P., Kumaraguru, P., 2022. Towards adversarial evaluations for inexact machine unlearning. arXiv .

Hershey, J.R., Olsen, P.A., 2007. Approximating the kullback leibler divergence between gaussian mixture models, in: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, IEEE. pp. IV–317.

Hu, Z., Zhang, Y., Xiao, M., Wang, W., Feng, F., He, X., 2024. Exact and efficient unlearning for large language model-based recommendation. Proceedings of ACM Conference .

Hunt, L., 2024. Critical problems that need to be solved at the intersection of ai, ip/data, copyright, and law (linkedin article). https://www.linkedin.com/posts/louiswhunt_i-recently-left-the-brilliant-team-at-liquid-activity\protect\discretionary{\char\hyphenchar\font}{}{}7273794644964950016-_sJO/. [Accessed 18-12-2024].

Jang, J., Yoon, D., Yang, S., Cha, S., Lee, M., Logeswaran, L., Seo, M., 2023. Knowledge unlearning for mitigating privacy risks in language models, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Toronto, Canada. pp. 14389–14408.

Department of Justice, C., 2024. California consumer privacy act (ccpa) state of california - department of justice - office of the attorney general. https://oag.ca.gov/privacy/ccpa. (Accessed on 03/08/2024).

Karamolegkou, A., Li, J., Zhou, L., Søgaard, A., 2023. Copyright violations and large language models, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore. pp. 7403–7412.

Kerrigan, G., Slack, D., Tuyls, J., 2020. Differentially private language models benefit from public pre-training, in: Proceedings of the Second Workshop on Privacy in NLP, Association for Computational Linguistics, Online. pp. 39–45.

Khashabi, D., Min, S., Khot, T., Sabharwal, A., Tafjord, O., Clark, P., Hajishirzi, H., 2020. UNIFIEDQA: Crossing format boundaries with a single QA system, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics. pp. 1896–1907.

Lev, O., Wilson, A., 2024. Faster machine unlearning via natural gradient descent. arXiv .

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L., 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics. pp. 7871–7880.

Li, H., Guo, D., Fan, W., Xu, M., Huang, J., Meng, F., Song, Y., 2023. Multi-step jailbreaking privacy attacks on ChatGPT, in: Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore. pp. 4138–4153.

Lin, C.Y., 2004. ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain. pp. 74–81.

Lin, J., 1991. Divergence measures based on the shannon entropy. IEEE Transactions on Information Theory 37, 145–151.

Lin, S., Hilton, J., Evans, O., 2022. TruthfulQA: Measuring how models mimic human falsehoods, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland. pp. 3214–3252.

Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al., 2024. Deepseek-v3 technical report. CoRR .

Liu, B., Liu, Q., Stone, P., 2022. Continual learning and private unlearning, in: Conference on Lifelong Learning Agents, PMLR. pp. 243–254.

Liu, S., Yao, Y., Jia, J., Casper, S., Baracaldo, N., Hase, P., Xu, X., Yao, Y., Li, H., Varshney, K.R., et al., 2025. Rethinking machine unlearning for large language models. Nature Machine Intelligence .

Lukas, N., Salem, A., Sim, R., Tople, S., Wutschitz, L., Zanella-Beguelin, S., 2023. Analyzing Leakage of Personally Identifiable Information in Language Models , in: 2023 IEEE Symposium on Security and Privacy (SP), IEEE Computer Society, Los Alamitos, CA, USA. pp. 346–363.

Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z.C., Kolter, J.Z., 2024. TOFU: A task of fictitious unlearning for LLMs. Secure and Trustworthy LLM Workshop, ICLR 2024 .

Malinka, K., Peresíni, M., Firc, A., Hujnák, O., Janus, F., 2023. On the educational impact of ChatGPT: Is artificial intelligence ready to obtain a university degree?, in: Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1, pp. 47–53.

Manzoor, A., Ziegler, S.C., Garcia, K.M.P., Jannach, D., 2024. Chatgpt as a conversational recommender system: A user-centric analysis, in: Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization, Association for Computing Machinery. p. 267–272.

Meisenbacher, S., Chevli, M., Vladika, J., Matthes, F., 2024. DP-MLM: Differentially private text rewriting using masked language models, in: Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand. pp. 9314–9328.

Nguyen, H.V., Vreeken, J., 2015. Non-parametric jensen-shannon divergence, in: Appice, A., Rodrigues, P.P., Santos Costa, V., Gama, J., Jorge, A., Soares, C. (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, Cham. pp. 173–189.

Panaretos, V.M., Zemel, Y., 2019. Statistical aspects of wasserstein distances. Annual review of statistics and its application 6, 405–431.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C., 2024. Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems 36.

Sánchez, D., Batet, M., 2016. C-sanitized: A privacy model for document redaction and sanitization. Journal of the Association for Information Science and Technology 67, 148–163.

Sweeney, L., 2002. k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems 10, 557 – 570.

The European Parliament, E., 2019. Directive - 2019/790 - en - dsm - eur-lex. http://data.europa.eu/eli/dir/2019/790/oj. (Accessed on 04/08/2024).

Tian, B., Liang, X., Cheng, S., Liu, Q., Wang, M., Sui, D., Chen, X., Chen, H., Zhang, N., 2024. To forget or not? towards practical knowledge unlearning for large language models, in: Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA. pp. 1524–1537.

Tirumala, K., Markosyan, A., Zettlemoyer, L., Aghajanyan, A., 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. Advances in Neural Information Processing Systems 35, 38274–38290.

Trippa, D., Campagnano, C., Bucarelli, M.S., Tolomei, G., Silvestri, F., 2024. Gradient-based and task-agnostic machine unlearning, in: Proceedings of the 41st International Conference on Machine Learning (ICML), PMLR. pp. –.

U.S. Copyright Office, U., 2018. Chapter 1 - circular 92 — u.s. copyright office. https://www.copyright.gov/title17/92chap1.html#106. (Accessed on 02/08/2024).

Voigt, P., Von dem Bussche, A., 2017. The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing 10, 10–5555.

Weidinger, L., Uesato, J., Rauh, M., Griffin, C., Huang, P.S., Mellor, J., Glaese, A., Cheng, M., Balle, B., Kasirzadeh, e.a., 2022. Taxonomy of risks posed by language models, in: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, Association for Computing Machinery, New York, NY, USA. p. 214–229.

Wu, T., Tao, C., Wang, J., Yang, R., Zhao, Z., Wong, N., 2025. Rethinking Kullback-Leibler divergence in knowledge distillation for large language models, in: Proceedings of the 31st International Conference on Computational Linguistics, Association for Computational Linguistics, Abu Dhabi, UAE. pp. 5737–5755.

Xu, H., Zhu, T., Zhang, L., Zhou, W., Yu, P.S., 2024. Machine unlearning: A survey. ACM Comput. Surv. 56.

Yao, J., Chien, E., Du, M., Niu, X., Wang, T., Cheng, Z., Yue, X., 2024. Machine unlearning of pre-trained large language models, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand. pp. 8403–8419.

Yao, Y., Xu, X., Liu, Y., 2023. Large language model unlearning. NeurIPS 2023 Workshop SoLaR .