READY2UNLEARN: A LEARNING-TIME APPROACH FOR PREPARING MODELS WITH FUTURE UNLEARNING READINESS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper introduces **Ready2Unlearn**, a learning-time optimization approach designed to facilitate future unlearning processes. Unlike the majority of existing unlearning efforts that focus on designing unlearning algorithms, which are typically implemented reactively when an unlearning request is made during the model deployment phase, Ready2Unlearn shifts the focus to the training phase, adopting a *"forward-looking"* perspective. Building upon well-established meta-learning principles, Ready2Unlearn proactively trains machine learning models with *unlearning readiness*, such that they are well prepared and can handle future unlearning requests in a more efficient and principled manner. Ready2Unlearn is model-agnostic and compatible with any gradient ascent-based machine unlearning algorithms. We evaluate the method on both vision and language tasks under various unlearning settings, including class-wise unlearning and random data unlearning. Experimental results show that by incorporating such preparedness at training time, Ready2Unlearn produces an *unlearning-ready* model state, which offers several key advantages when future unlearning is requested, including reduced unlearning time, improved retention of overall model capability, and enhanced resistance to the inadvertent recovery of forgotten data. We hope this study could inspire future work to explore more proactive strategies for equipping machine learning models with built-in readiness towards more reliable and principled machine unlearning.

1 INTRODUCTION

Machine unlearning (Cao & Yang, 2015) refers to the process of removing the imprint left by specific data samples during the training of a machine learning model. AI developers employ machine unlearning for various purposes. In the context of privacy protection, it is often necessary to remove the influence that individuals' personal data has had on a model's learned parameters (Jang et al., 2023; Miranda et al., 2024; Zhou et al., 2023). Legal frameworks such as the European Union's General Data Protection Regulation (GDPR) (Protection, 2018) and the California Consumer Privacy Act (CCPA)[1] grant individuals the right to control their personal data, including revoking it from organizations that use the data to train models. Beyond privacy, machine unlearning is also used to address ethical and security concerns by removing the influence of harmful or sensitive data (Schoepf et al., 2024; Yu et al., 2023), such as preventing large language models (LLMs) from retaining information that could be misused for developing bioweapons or launching cyberattacks (Barrett et al., 2023; Sandbrink, 2023; Li et al., 2024). Additionally, unlearning can be applied to improve model performance by eliminating the impact of low-quality or noisy training samples (Cao & Yang, 2015; Wang et al., 2023). These diverse applications emphasize that machine unlearning is important and practically meaningful.

Numerous unlearning algorithms have been introduced in recent years, employing techniques such as preference optimization (Zhang et al., 2024b; Mekala et al., 2025), gradient rectification (Hoang et al., 2024; Lin et al., 2024; Wang et al., 2025b), and data augmentation (Peng et al., 2025; Cha et al., 2024; Mekala et al., 2025), to name a few. Despite these efforts, unlearning remains a chal-

[1] `https://oag.ca.gov/privacy/ccpa`

lenging task. First, it often requires considerable time or a large number of optimization steps to achieve satisfactory forgetting, especially for large-scale models such as LLMs (Eldan & Russinovich, 2023). Additionally, balancing the trade-off between forgetting data and preserving overall model utility is difficult (Hoang et al., 2024; Lin et al., 2024; Wang et al., 2025b), as the unlearning process can lead to catastrophic forgetting (Zhang et al., 2024a). Moreover, some studies suggest that current unlearning methods may not be as reliable as they appear, with data that seems to be "forgotten" often being easily recovered (Kim et al., 2025; Zhang et al., 2024c; Hong et al., 2024).

Such long-standing challenges prompt us to ask: *Is the model truly ready to forget when unlearning is initiated, and can we take steps during training to proactively prepare it with unlearning readiness against potential future unlearning requests?* In this paper, we explore the possibility that equipping the model with unlearning readiness during the training phase to benefit the unlearning process that may take place later after model deployment, with improved efficiency and reliability.

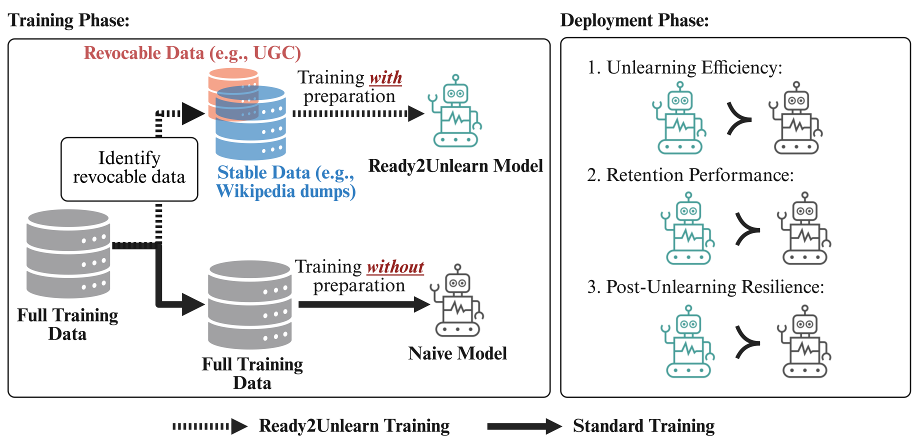

Figure 1: Comparison of learning with (top) and without (bottom) unlearning preparation.

The problem is depicted in Figure 1. We make a practical assumption that, in real-world applications, not all training data is equally likely to be subject to future unlearning requests (Krishnan et al., 2025). Some data, such as user-generated content (UGC) (De Min et al.), are more likely to be revoked due to privacy or other regulatory concerns, while other data, like public datasets (e.g., Wikipedia dumps), are less likely to trigger such requests.

Motivated by this observation, we introduce a method (Ready2Unlearn) that equips the model, during training, with an awareness of potential unlearning actions that may occur later during deployment for data that are more likely to be revoked, which we term *revocable data* throughout the manuscript. The remaining data are therefore termed *stable data*, reflecting their inherent stability, that is, their low likelihood of being subject to future revocation requests. In terms of which data should be classified as revocable or stable, we consider this to be largely a managerial decision, guided by organizational priorities and regulatory considerations.

Our expectation is that, once a model has undergone learning with Ready2Unlearn and is deployed in the environment, and an unlearning request arises for the revocable data, it can 1) unlearn more efficiently with fewer unlearning steps, 2) better preserve performance on the unaffected stable data, and 3) exhibit greater post-unlearning robustness against the inadvertent recovery of *previously forgotten data*, compared to a model trained without unlearning readiness (the naïve model). Two points to note here: First, by "learning", we refer to either training from scratch or further fine-tuning (as commonly done in the LLM context). Second, by "unlearning", we mean any gradient ascent-based unlearning algorithms.

Ready2Unlearn adopts meta-learning principles, particularly the MAML algorithm (Finn et al., 2017). Traditional meta-learning techniques aim to find a model initialization such that performing a small number of gradient *descent* steps from this starting point allows the model to quickly adapt to new tasks. In our setting, we aim to obtain a model "initialization" (the prepared model state) such that, based on this initialization, applying only a few gradient *ascent* steps (representing unlearning operations) leads to a significant increase in loss for the intended forgotten data, while preserving overall model utility and ensuring strong post-unlearning resilience.

To demonstrate preparing models for unlearning readiness at the learning stage is feasible, we test the proposed Ready2Unlearn idea on both vision and language unlearning tasks, including class-wise unlearning and random data unlearning. In our experiments, we show that with Ready2Unlearn preparation, the model responds much more quickly to unlearning requests compared to models without such preparedness or baseline approaches. Additionally, the model's overall utility is better preserved, with reduced risk of catastrophic forgetting. Furthermore, when attempting to recover the forgotten data by further fine-tuning the unlearned model on data with similar distributional characteristics, such as stylistically or semantically related examples, the model exhibits greater resistance to recovery compared to its unprepared counterpart. We hope our findings offer fresh perspectives in machine unlearning and additionally inspire future work on incorporating such "looking-ahead" designs in broader algorithmic contexts.

2 Related Work

Machine unlearning. Due to the growing need to eliminate traces of specific data from machine learning models throughout the AI lifecycle, machine unlearning techniques have seen rapid advancement (Liu et al., 2024b; Wang et al., 2024a). Current research in this area primarily focuses on developing effective unlearning algorithms (Peng et al., 2025; Lin et al., 2024; Jia et al., 2024; Ji et al., 2024; Zhou et al., 2023; Jang et al., 2023; Fan et al., 2023; Chen & Yang, 2023), designing rigorous evaluation protocols (Liu et al., 2025; Kim et al., 2025; Zhang et al., 2024c; Shi et al., 2024; Hong et al., 2024), and identifying and addressing practical, real-world challenges (Shen et al., 2024), such as the unavailability of users' erased data during the unlearning process (Wang et al., 2025a), which relaxes some classic assumptions in traditional unlearning practices. While existing machine unlearning methods generally perform well in typical scenarios, they often face critical limitations, such as prolonged unlearning time (Eldan & Russinovich, 2023), catastrophic forgetting (Peng et al., 2025; Wang et al., 2025b; Lin et al., 2024; Hoang et al., 2024; Choi et al., 2024), and vulnerability under more rigorous evaluation conditions, such as susceptibility to jailbreak attacks (Zhao et al., 2024a) and the ease with which forgotten data can be recovered (Kim et al., 2025; Hong et al., 2024; Zhang et al., 2024c). This suggests that current unlearning solutions are inadequate and still have considerable room for improvement. Our work contributes to this line of research by demonstrating that it is possible to prepare models at the learning stage with future unlearning readiness to further enhance the efficiency, reliability, and robustness of existing unlearning methods. We believe this offers a new perspective for enhancing current unlearning practices.

Training-time regularization to mitigate memorization. Mitigating training data memorization is a reasonable approach to alleviate future unlearning efforts, as models that memorize less are less tied to specific details of data points, making it easier to forget them later (Zhao et al., 2024b). This is typically achieved through regularization techniques applied during model training. Well-known examples include dropout (Srivastava et al., 2014), weight decay (Ng, 2004; Krogh & Hertz, 1991), data augmentation (Shorten & Khoshgoftaar, 2019), and differentially private learning (Abadi et al., 2016), to name a few. In addition to these general, model-agnostic methods, there are more specialized techniques, such as Goldfish regularization (Hans et al., 2024), which randomly excludes tokens from loss computation, and NEFTune (Jain et al., 2023), which adds noise to embedding vectors, both tailored for large language models. At a broader level, the Ready2Unlearn method presented in this work can also be positioned within the regularization literature. Furthermore, it adds to this line of research with a novel unlearning-specific regularization technique. Ready2Unlearn is particularly advantageous in machine unlearning scenarios, where existing general regularization techniques often fall short, as they are, by nature, not optimized for unlearning contexts. Thus, we believe this work provides a more targeted solution from this perspective.

Meta-learning. Another relevant line of research is meta-learning, also known as "learning to learn", which aims to train models that can rapidly adapt to new tasks with limited data or computational resources (Hospedales et al., 2021). A notable example is model-agnostic meta-learning (MAML) (Finn et al., 2017), which seeks to find a model initialization such that only a few gradient updates are required for effective adaptation to new tasks. This paradigm has been widely adopted across various fields, such as domain generalization (Li et al., 2018), safeguarding LLMs against adversarial attacks (Tamirisa et al., 2024), and hyperparameter optimization (Baik et al., 2020). Our work draws inspiration from the core idea of MAML, but shifts the objective from fast adaptation to new tasks to fast, reliable machine unlearning. By incorporating a meta-objective at training time, Ready2Unlearn optimizes the model into an *unlearning-ready* state—a parameter configuration from which future gradient ascent updates (representing unlearning) can proceed in a well-behaved and principled manner. This preparation leads to several desirable properties, including improved unlearning efficiency, better retention of overall model capability, and increased resistance to the reintroduction of forgotten data. We are aware of prior work that also applies meta-learning techniques in the context of machine unlearning (Huang et al., 2024). However, our approach differs fundamentally in both the timing and the goal of applying meta-learning: their method adopts meta-learning during the unlearning phase, after model deployment, whereas Ready2Unlearn introduces meta-learning at training time, proactively preparing the model before any unlearning request arises. This distinction positions our approach as a *preemptive* strategy, marking a conceptual departure from the majority of reactive unlearning methods toward a proactive paradigm. In this sense, we believe Ready2Unlearn represents a novel extension of meta-learning principles into the field of machine unlearning.

3 Learning with Unlearning Preparedness

3.1 Unlearning Process

We assume that the model developer has access to a dataset $\mathcal{D} = \mathcal{D}_{\mathrm{f}} \cup \mathcal{D}_{\mathrm{r}}$, which comprises both revocable data (likely to be unlearned in the future, referred to as *forget data*[2], $\mathcal{D}_{\mathrm{f}}$) and stable data (unlikely to be unlearned, referred to as *retain data*, $\mathcal{D}_{\mathrm{r}}$), and builds a model with weights θ_P, where a preparation $\mathtt{P}$ has been applied. Our goal is to design $\mathtt{P}$ such that θ_P performs well on three metrics when future unlearning is triggered: $\mathtt{efficiency\_metric}(\theta_P)$, $\mathtt{retention\_metric}(\theta_P)$, and $\mathtt{resistance\_metric}(\theta_P)$. In this work, by "unlearning", we refer to the process of applying gradient ascent steps to adjust θ_P based on the forget data. Gradient ascent is widely used as an effective unlearning strategy due to its simplicity and its model- and data-agnostic nature (Yao et al., 2023; Maini et al., 2024; Graves et al., 2021; Hoang et al., 2024; Li et al., 2023; Tarun et al., 2023; Thudi et al., 2022; Ji et al., 2024). Moreover, we assume that the retain data is not accessible during unlearning. We impose this stricter condition because, in many real-world scenarios, access to retain data is often impossible, and retraining the model on this data can be prohibitively costly (Wang et al., 2024b; Cheng et al., 2024; Foster et al., 2024). Thus, throughout the paper, unlearning specifically refers to a clean process where gradient ascent steps are applied solely to the forget data.

3.2 Problem Formulation and Metrics

Let $\mathtt{GA}$ denote the gradient ascent unlearning operation, which maps the prepared model θ_P to the *unlearned model* $\theta'_P = \mathtt{GA}(\theta_P; \mathcal{D}_{\mathrm{f}})$. Let $\mathtt{RC}$ denote the recovery operation, which further fine-tunes the unlearned model θ'_P on data samples $\mathcal{D}_{\mathrm{rc}}$, similar in style to the forget data, producing the *post-recovery model* $\theta''_P = \mathtt{RC}(\theta'_P; \mathcal{D}_{\mathrm{rc}})$. Below, we define three key metrics.

Efficiency metric. We say a preparation $\mathtt{P}$ leads to efficient unlearning if the model θ_P experiences a substantial increase in loss on the forget data after only a few, or even a single, gradient ascent unlearning update. Thus, $\mathtt{efficiency\_metric}$ is defined as the loss (e.g., classification error) of the unlearned model on the forget data. A higher loss indicates greater unlearning efficiency.

Retention metric. We say a preparation $\mathtt{P}$ enables strong capability retention if the unlearned model θ'_P preserves much of its performance on the retain data. Thus, $\mathtt{retention\_metric}$ is defined as the unlearned model's performance (e.g., classification accuracy) on the retain data. Higher performance signifies stronger retention.

Resistance metric. We say a preparation $\mathtt{P}$ equips the model with greater resistance to the inadvertent recovery of erased data if, upon further fine-tuning the unlearned model on data similar in style to the forget data, the resulting model is less likely to regain information about the forgotten data.

[2] Here, in presenting the methodology, we use the terms "forget data" and "retain data" to denote revocable and stable data, respectively, to align with the conventional terminology in the machine unlearning literature.

Thus, `resistance_metric` is defined as the loss of the post-recovery model θ''_P on the forget data. A higher loss indicates stronger post-unlearning resilience.

3.3 UNLEARNING-READY TRAINING

To proactively prepare models during learning towards more efficient and principled unlearning in the future, we introduce Ready2Unlearn, a *forward-looking* method outlined in Algorithm 1 in Appendix A. Inspired by meta-learning, this approach prepares the model during learning to be *unlearning-ready*, ensuring that future unlearning via gradient ascent behaves in a stable and reliable manner. At a high level, we learn the model θ_P with unlearning preparedness using a *dual-loop* optimization structure inspired by MAML (Finn et al., 2017), comprising an *inner-loop* gradient update and an *outer-loop* optimization.

Method intuition. The key idea behind Ready2Unlearn is to *optimize for the future*. Rather than maximizing the model's immediate performance, we simulate potential unlearning operations that may occur later and optimize the model to be ready for them. Specifically, this is achieved by designing the inner-loop gradient update to *mimic* unlearning actions, representing the "unlearner's" first move. In the outer loop, the model parameters are optimized to maximize three desirable properties—efficiency, retention, and resistance—against the unlearner's move simulated in the inner loop. This forward-looking optimization ensures that, when unlearning is eventually triggered, the model will exhibit optimal performance in terms of unlearning efficiency, capability retention, and post-unlearning resilience. Essentially, we are preparing the model for future unlearning challenges rather than simply optimizing for the current task. This forward-looking perspective distinguishes our method from conventional approaches. We illustrate this forward-looking nature using conceptual 1D loss landscapes in Figure 2.

Unlearning-ready objective. Let $\mathcal{L}$ denote the loss function suitable for the task at hand. We now describe the optimization objective of Ready2Unlearn.

To *enhance unlearning efficiency* (i.e., to achieve fast unlearning), we seek to find a θ upon which even a single step of gradient ascent (representing an unlearning step), $\texttt{GA}(\theta)$, leads to a substantial increase in loss on the forget data. Thus, we consider maximizing $\mathcal{L}(\texttt{GA}(\theta); \mathcal{D}_\text{f})$.

To *avoid catastrophic forgetting* (i.e., to promote performance retention), we aim to identify a θ such that, after unlearning, the resulting parameters, $\texttt{GA}(\theta)$, still enable the model to preserve much of its performance on the retain data. Accordingly, we consider minimizing $\mathcal{L}(\texttt{GA}(\theta); \mathcal{D}_\text{r})$.

To *achieve more reliable unlearning* (i.e., to improve post-unlearning resilience), we aim to find a θ such that, after unlearning with $\texttt{GA}(\theta)$ and further fine-tuning the unlearned model on recovery data (which shares a similar style to the forget data), the resulting model does not regain significant information about the forgotten data. We operationalize this by minimizing $\mathcal{L}(\texttt{GA}(\theta); \mathcal{D}_\text{rc})$. The rationale behind is to direct the unlearning process so that it removes the most distinctive characteristics of the forget data, rather than merely erasing superficial patterns that may also be present in the recovery data. Otherwise, if the model only unlearns superficial patterns, the loss on the recovery data is likely to increase as well, which is exactly what we aim to prevent by minimizing the aforementioned loss.

Put together, our objective is to solve the following optimization problem to obtain the unlearning-ready model θ_P:

$$\min_\theta [-\mathcal{L}(\texttt{GA}(\theta), \mathcal{D}_\text{f}) + \lambda_1 \cdot \mathcal{L}(\texttt{GA}(\theta), \mathcal{D}_\text{r}) + \lambda_2 \cdot \mathcal{L}(\texttt{GA}(\theta), \mathcal{D}_\text{rc}) + \lambda_3 \cdot \mathcal{L}(\theta; \mathcal{D}))], \tag{1}$$

where λ_1, λ_2, and λ_3 are scalar weights for the respective losses. The first three terms represent the *"future objectives"*, ensuring that when future unlearning takes place, their respective objectives will be optimized accordingly, reflecting the forward-looking nature of Ready2Unlearn. The final term serves as the *"current objective"*, optimizing the model's current utility prior to any unlearning action being taken. Please refer to Algorithm 1 in Appendix A for the full optimization procedure.

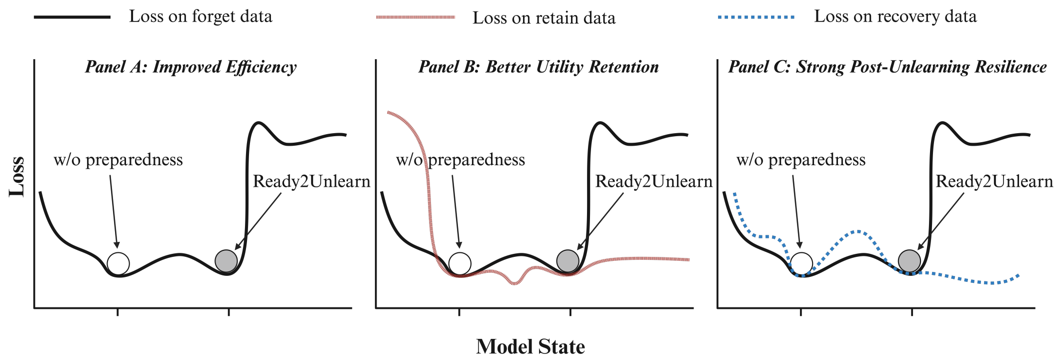

Figure 2: An illustration of the forward-looking nature of Ready2Unlearn using conceptual 1D loss landscapes. **Panel A:** The model state obtained with unlearning preparedness (gray circle) lies adjacent to a steep ascent in the loss landscape with respect to the forget data, such that even a single gradient ascent step can trigger a substantial loss increase, enabling fast and effective unlearning. In contrast, the unprepared model (white circle) lies in a flatter region far from any steep increase, requiring many steps to achieve comparable unlearning. **Panel B:** The model state optimized by Ready2Unlearn resides in a region where the retain-data loss remains low and stable despite gradient ascent updates on forget data, thereby preserving the model's utility. The unprepared model resides in a region where unlearning actions (i.e., gradient ascent on forget data) adversely impact performance on retain data, as indicated by a sharp increase in retain loss. **Panel C:** Around the unprepared model state, the loss landscapes of forget data and recovery data exhibit high similarity. As a result, fine-tuning on recovery data inadvertently lowers the forget loss as well, undoing the unlearning. In contrast, Ready2Unlearn prepares the model in a region where the recovery loss is low and exhibits a distinct pattern from the forget loss, thereby making the model less likely to re-acquire forgotten information during further fine-tuning on recovery data.

4 Experiments

4.1 Class-Wise Unlearning in Image Classification

Experiment setup. We consider image classification tasks using the MNIST (Deng, 2012) and PathMNIST (Yang et al., 2023; 2021) datasets. Each class is treated in turn as the revocable data (i.e., forget data, $\mathcal{D}_{\mathrm{f}}$), prepared at training time for potential future unlearning requests. The remaining classes are treated as stable data (i.e., retain data, $\mathcal{D}_{\mathrm{r}}$), used to evaluate the model's capability retention after unlearning. Following prior unlearning research (Zhou et al., 2023), we use a convolutional neural network (CNN) as the classifier. The model is trained using the negative log-likelihood loss function, and performance is evaluated using classification accuracy. We apply a first-order approximation to compute the meta-gradients (as detailed in lines 6, 7, and 8 in Algorithm 1). We use an inner-loop unlearning rate α of 1×10^{-5}, an outer-loop learning rate η of 2×10^{-4}, and set the loss weights λ_1 and λ_3 to 2 and 4, respectively, during training. In the deployment phase, when unlearning is executed, we apply gradient ascent on the forget data with a step size of 1×10^{-5}.

Baseline methods. We consider several baseline methods that reduce the training imprint of revocable data, enabling more efficient future unlearning. *Standard Training* serves as the basic approach, where the model is trained using SGD without distinguishing between revocable and stable data. *Loss Reweighting* reduces the influence of revocable data during training by assigning it half the loss weight of stable data (Kendall et al., 2018). *Noisy Training* perturbs revocable images by adding standard Gaussian noise scaled by 0.3, which helps to prevent the model from overly memorizing these examples and supports easier unlearning in the future (Rifai et al., 2011; Shorten & Khoshgoftaar, 2019). In *Clipped Training*, gradient clipping is applied to revocable data to constrain its influence on model parameters, which may help prevent overfitting and support easier unlearning (Mikolov et al., 2012). Finally, *Phased Training* starts with training on the full dataset in the first half of the training period and then proceeds to train only on stable data in the second half, allowing the model to initially learn from revocable data without continued exposure, which may ease future unlearning (Bengio et al., 2009; Goodfellow et al., 2013).

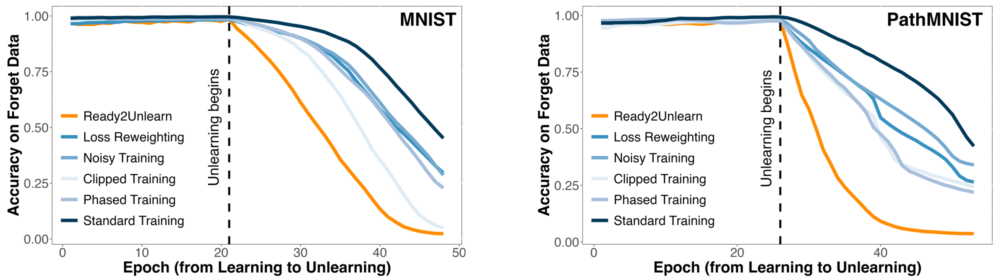

Figure 3: Comparison of unlearning efficiency for MNIST (left) and PathMNIST (right). Each line represents the average forget-data accuracy across all class-wise unlearning settings, where each class is treated as the forget class in turn. All methods are evaluated with the same unlearning rate of 1×10^{-5} for a fair comparison. The vertical dashed line marks the moment when unlearning begins.

Results and discussion. We present the unlearning efficiency benchmarking results in Figure 3, which lead to several key observations. First, when the model is trained without consideration for future unlearning, the unlearning process is notably slow. This is evident from the Standard Training baseline, where the accuracy on the forget data remains consistently higher than that of all other methods, which incorporate varying degrees of unlearning preparedness. This underscores the importance of training-time preparation for enabling more efficient unlearning later. Second, among the evaluated methods, Ready2Unlearn exhibits the highest unlearning efficiency. We observe that once unlearning begins, the model prepared with Ready2Unlearn immediately undergoes a sharp decline in accuracy on the forget data, reflecting a quick response compared to other models. For example, when the model equipped with Ready2Unlearn reaches 50% accuracy, the other models still maintain a relatively high accuracy of around 75% on average. Third, at the moment when unlearning is initiated (marked by the vertical dashed line), the accuracies on the forget data are comparable across all methods, indicating that the superior efficiency of Ready2Unlearn is not at the cost of suboptimal performance on the forget data prior to unlearning. In other words, the unlearning readiness enabled by our approach does not significantly compromise the model's pre-unlearning performance. Overall, the results demonstrate that training a model with unlearning in mind improves the unlearning process at a later stage, with more tailored approaches, such as Ready2Unlearn, offering greater advantages. We also examine how the duration of preparatory steps with Ready2Unlearn affects future unlearning efficiency; please see Appendix C for details.

We evaluate whether Ready2Unlearn equips the model with better capability retention when unlearning is performed. Specifically, we examine the model's accuracy on the retain data after unlearning has driven the forget class accuracy down to the level of random guessing. The results visualized in Figure 4 reveal that models trained with Ready2Unlearn consistently maintain substantially higher accuracy on retain data compared to those trained without unlearning preparation (i.e., Standard Training). It is important to note that during unlearning, we assume retain data is inaccessible; thus, unlearning is performed solely by applying gradient ascent to the forget data, without any concurrent training on retain data as is often done in prior work (Huang et al., 2024). Thus, the improved retention of performance is entirely attributed to the training-time preparation, emphasizing the advantage of Ready2Unlearn in handling more complex unlearning scenarios where the retain data is inaccessible, with a forward-looking design.

4.2 Random Data Unlearning in Text Generation

Experiment setup. We use LLaMA-3.2-1B[3] as the target model and benchmark unlearning efficiency on two widely adopted unlearning corpora: MUSE-Books and MUSE-News (Shi et al., 2024). We use their "raw" data splits and adhere to the official forget/retain partitions for both datasets. To evaluate whether Ready2Unlearn improves post-unlearning resilience, we use a separate dataset, the Enron email corpus[4], which offers an ideal setting for recovery evaluation due to the strong stylistic consistency across email messages. We randomly split the data into three subsets:

[3]https://huggingface.co/meta-llama/Llama-3.2-1B
[4]https://www.cs.cmu.edu/~enron/

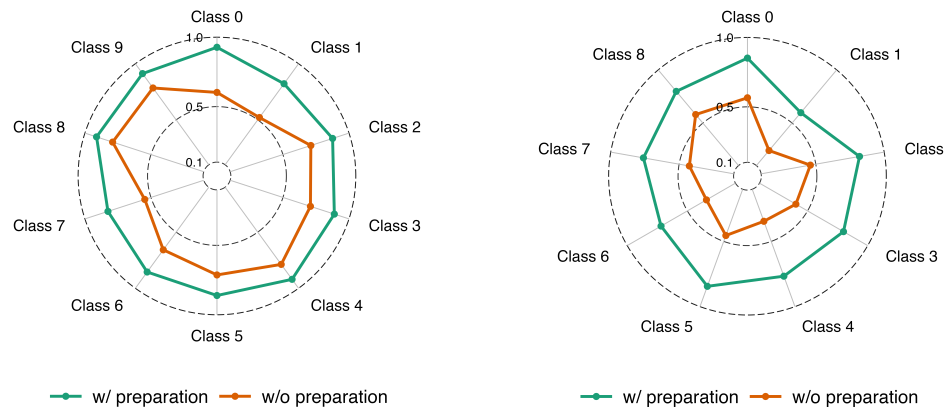

Figure 4: Performance retention for MNIST (left) and PathMNIST (right). Each axis of the radar chart corresponds to a class treated as the forget class. The value on each axis shows the model's retain accuracy when its forget accuracy reaches random guessing.

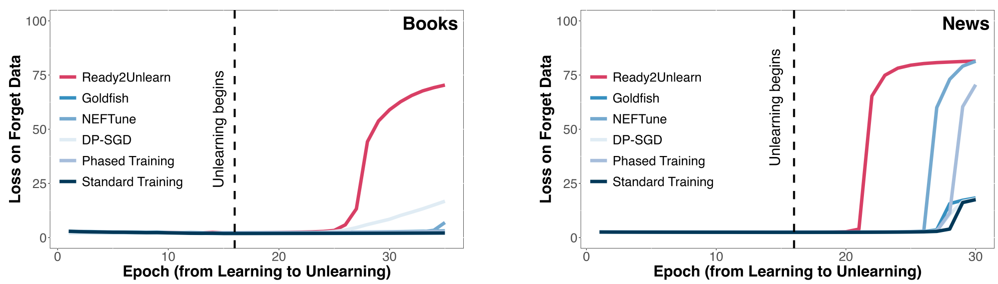

Figure 5: Comparison of unlearning efficiency for MUSE-Books (left) and MUSE-News (right) using Llama-3.2-1B as the target model. Each line represents the cross-entropy loss on the forget data for each method. All methods are evaluated with the same unlearning rate of 1×10^{-6} for a fair comparison. The vertical dashed line marks the moment when unlearning begins.

one for $\mathcal{D}_{\text{f}}$, one for $\mathcal{D}_{\text{r}}$, and one ($\mathcal{D}_{\text{rc}}$) for further fine-tuning the unlearned model to assess potential recovery of forgotten information. Throughout the experiments, by learning, unlearning, and recovering, we refer to fine-tuning the model with the next-word prediction objective using cross-entropy loss (also used as the performance evaluation metric). For unlearning efficiency evaluation, we set the loss scaling factors λ_1 and λ_3 to 2 and 4, respectively. For post-unlearning resilience evaluation, we set $\lambda_2 = 3$ and $\lambda_3 = 4$. We apply gradient ascent unlearning with a step size of 1×10^{-6}.

Baseline methods. We consider the following baseline methods. *Standard Training* and *Phased Training* are included as defined in the earlier class-wise unlearning setup. We additionally consider *DP-SGD* (Li et al., 2021; Abadi et al., 2016), which fine-tunes the model with differential privacy using a clipping norm of 0.1. We include two language model-specific techniques: *Goldfish* (Hans et al., 2024), which randomly excludes tokens from the loss computation with a probability of 0.25, and *NEFTune* (Jain et al., 2023), which injects noise into the embedding vectors with a scaling factor of $\alpha = 5$. These techniques are applied to revocable data during training to mitigate over-memorization, thereby enabling more efficient unlearning later on.

Results and discussion. We compare the unlearning efficiency of all methods in Figure 5. The results indicate that with a very small gradient ascent step size during unlearning (as in this experiment, 1×10^{-6}), achieving sufficient forgetting for LLMs typically necessitates a considerable number of optimization steps—meaning that the effects of unlearning are not immediately apparent upon initiation. Fortunately, consistent with the findings from the class-wise unlearning experiments, incorporating preparatory steps during training can effectively reduce the time needed to achieve meaningful unlearning later on, without significantly compromising the model's performance prior to unlearning. Notably, models trained with Ready2Unlearn begin to forget earlier than all baselines and reach noticeable unlearning with fewer gradient ascent steps, demonstrating that our approach

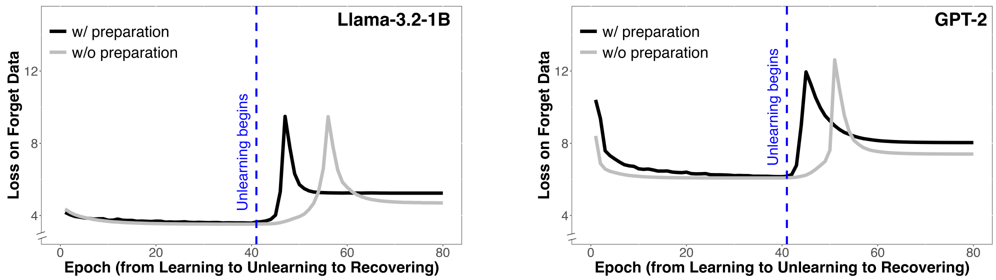

Figure 6: Loss on forget data across three phases for Llama-3.2-1B (left) and GPT-2 (right). The loss decreases and plateaus during training, rises sharply during unlearning, and then decreases and plateaus again during recovery via further fine-tuning on stylistically similar data.

`1. login: pallen pw: ke9davis I do not think these are required by the ISP 2. static IP address`

`1. login: pallen pw: ke9davis I do not think these are required by the ISP 2. static IP address`

Figure 7: A visual example from the forget set comparing token-level loss after unlearning a model trained without preparation (Standard Training, top) and with Ready2Unlearn preparation (bottom). Color shading indicates relative loss per token, with darker tones representing higher values. In the standard case, loss is spread more uniformly across tokens, suggesting that the model treats all content equally during unlearning rather than prioritizing the removal of more critical information. In contrast, the model prepared by Ready2Unlearn tends to assign higher loss to more distinctive, data-specific tokens, such as login names or passwords (e.g., "pallen", "ke9davis"), showcasing its focus on unlearning more meaningful, sensitive information rather than generic content. This makes the removed information harder to recover without access to the original forget data.

generalizes well to transformer-based language models, beyond classic deep neural networks like CNNs. Additional evaluation results using Llama-3.2-3B[5], GPT-2, and GPT-2 Medium (Radford et al., 2019) as target models are provided in Appendix B.

Figure 6 compares the post-unlearning resilience of models trained with and without Ready2Unlearn preparation. The most notable observation is that, after the loss has plateaued following further fine-tuning the unlearned model on a dataset similar in style to the forget data, the model prepared with Ready2Unlearn consistently maintains a higher loss on the forget data compared to the model without preparation. This indicates that the Ready2Unlearn-prepared model is more resistant to regaining the forgotten information—even when exposed to data with similar characteristics—than its non-prepared counterpart. This resilience arises from the gradient term in line 8 in Algorithm 1 in Appendix A, which directs the model to unlearn the most distinctive, data-specific features of the forget data, rather than merely suppressing superficial patterns that could easily re-emerge during subsequent (inadvertent) fine-tuning. In contrast, without this preparatory step, the model is more susceptible to relearning those superficial patterns, leading to a lower loss on the forget data after fine-tuning. See Figure 7 for an illustration. Thus, from this perspective, we believe Ready2Unlearn generates new insights into more targeted machine unlearning (Liu et al., 2024a), where the information to be removed from the model is much more nuanced and selective.

5 Conclusion

In this paper, we introduce Ready2Unlearn, a forward-looking approach that proactively prepares neural network models during training to enhance their readiness for future unlearning. By incorporating preemptive steps into the learning process, our method enables models to unlearn more efficiently while preserving much of their utility and exhibiting greater resilience after unlearning. We demonstrate that unlearning should not only be treated as an afterthought but rather as a critical aspect of model lifecycle management that can be proactively addressed through forward-looking training designs. We believe this work offers a new perspective for addressing challenges posed by evolving data governance and privacy demands, particularly in the context of machine unlearning.

[5] https://huggingface.co/meta-llama/Llama-3.2-3B

6 LIMITATIONS

Our work also has limitations that can be improved in future research. First, our study simplifies the revocation risk categorization of training data into two groups: revocable data and stable data. This binary classification may not capture the full spectrum of revocation risks. Future work could explore more granular risk levels or even continuous risk gradations in training data, moving beyond the basic dichotomy. Second, our method relies on gradient-ascent-based unlearning algorithms. While gradient-ascent is predominantly used and effective in many cases, there are unlearning situations where it may not perform optimally. Future work could expand on our approach and generalize it to a broader range of unlearning algorithms, potentially addressing situations where gradient-ascent may fall short. Third, our method incorporates additional training objectives towards achieving future optimality, which may lead to a trade-off in the model's current performance. Although this trade-off is not significant in our experiments, it could become more pronounced in larger and more complex training scenarios. Future studies could explore ways to mitigate this trade-off, such as identifying key factors that influence the balance between current model performance and future unlearning preparedness, and finding conditions under which such compromises can be minimized. Finally, while we focus on the MAML training paradigm in this study, we acknowledge that other meta-learning algorithms could also be useful in this context. Future research could explore more advanced meta-learning techniques to enable more efficient preparation, such as reducing the preparatory duration required to achieve meaningful unlearning readiness.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Sungyong Baik, Myungsub Choi, Janghoon Choi, Heewon Kim, and Kyoung Mu Lee. Meta-learning with adaptive hyperparameters. *Advances in neural information processing systems*, 33:20755–20765, 2020.

Clark Barrett, Brad Boyd, Elie Bursztein, Nicholas Carlini, Brad Chen, Jihye Choi, Amrita Roy Chowdhury, Mihai Christodorescu, Anupam Datta, Soheil Feizi, et al. Identifying and mitigating the security risks of generative ai. *Foundations and Trends® in Privacy and Security*, 6(1):1–52, 2023.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pp. 463–480. IEEE, 2015.

Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 11186–11194, 2024. Issue: 10.

Jiaao Chen and Diyi Yang. Unlearn What You Want to Forget: Efficient Unlearning for LLMs. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12041–12052, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.738. URL `https://aclanthology.org/2023.emnlp-main.738`.

Xinwen Cheng, Zhehao Huang, Wenxin Zhou, Zhengbao He, Ruikai Yang, Yingwen Wu, and Xiaolin Huang. Remaining-data-free machine unlearning by suppressing sample contribution. *arXiv preprint arXiv:2402.15109*, 2024.

Dasol Choi, Soora Choi, Eunsun Lee, Jinwoo Seo, and Dongbin Na. Towards efficient machine unlearning with data augmentation: Guided loss-increasing (gli) to prevent the catastrophic model utility drop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 93–102, 2024.

Thomas De Min, Massimiliano Mancini, Stéphane Lathuilière, Subhankar Roy, and Elisa Ricci. Unlearning personal data from a single image. *Transactions on Machine Learning Research*.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

Ronen Eldan and Mark Russinovich. Who's Harry Potter? Approximate Unlearning in LLMs. *arXiv preprint arXiv:2310.02238*, 2023.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. SalUn: Empowering Machine Unlearning via Gradient-based Weight Saliency in Both Image Classification and Generation. In *The Twelfth International Conference on Learning Representations*, 2023.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 12043–12051, 2024. Issue: 11.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11516–11524, 2021. Issue: 13.

Abhimanyu Hans, Yuxin Wen, Neel Jain, John Kirchenbauer, Hamid Kazemi, Prajwal Singhania, Siddharth Singh, Gowthami Somepalli, Jonas Geiping, Abhinav Bhatele, and others. Be like a Goldfish, Don't Memorize! Mitigating Memorization in Generative LLMs. *arXiv preprint arXiv:2406.10209*, 2024.

Tuan Hoang, Santu Rana, Sunil Gupta, and Svetha Venkatesh. Learn to unlearn for deep neural networks: Minimizing unlearning interference with gradient projection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4819–4828, 2024.

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

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9): 5149–5169, 2021.

Mark He Huang, Lin Geng Foo, and Jun Liu. Learning to unlearn for robust machine unlearning. In *European Conference on Computer Vision*, pp. 202–219. Springer, 2024.

Neel Jain, Ping-yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, and others. Neftune: Noisy embeddings improve instruction finetuning. *arXiv preprint arXiv:2310.05914*, 2023.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge Unlearning for Mitigating Privacy Risks in Language Models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.805. URL https://aclanthology.org/2023.acl-long.805.

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

Jinghan Jia, Yihua Zhang, Yimeng Zhang, Jiancheng Liu, Bharat Runwal, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. SOUL: Unlocking the Power of Second-Order Optimization for LLM Unlearning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4276–4292, 2024.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

Yongwoo Kim, Sungmin Cha, and Donghyun Kim. Are we truly forgetting? a critical re-examination of machine unlearning evaluation protocols. *arXiv preprint arXiv:2503.06991*, 2025.

Aravind Krishnan, Siva Reddy, and Marius Mosbach. Not all data are unlearned equally. *arXiv preprint arXiv:2504.05058*, 2025.

Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Guanghao Li, Li Shen, Yan Sun, Yue Hu, Han Hu, and Dacheng Tao. Subspace based federated unlearning. *arXiv preprint arXiv:2302.12448*, 2023.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. In *International Conference on Machine Learning*, pp. 28525–28550. PMLR, 2024.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.

Shen Lin, Xiaoyu Zhang, Willy Susilo, Xiaofeng Chen, and Jun Liu. GDR-GMA: Machine Unlearning via Direction-Rectified and Magnitude-Adjusted Gradients. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 9087–9095, 2024.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, and others. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pp. 1–14, 2025. Publisher: Nature Publishing Group UK London.

Yujian Liu, Yang Zhang, Tommi Jaakkola, and Shiyu Chang. Revisiting Who's Harry Potter: Towards Targeted Unlearning from a Causal Intervention Perspective. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8708–8731, 2024a.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Machine unlearning in generative ai: A survey. *arXiv preprint arXiv:2407.20516*, 2024b.

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

Anmol Reddy Mekala, Vineeth Dorna, Shreya Dubey, Abhishek Lalwani, David Koleczek, Mukund Rungta, Sadid A Hasan, and Elita AA Lobo. Alternate Preference Optimization for Unlearning Factual Knowledge in Large Language Models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 3732–3752, 2025.

Tomáš Mikolov et al. Statistical language models based on neural networks. *Presentation at Google, Mountain View, 2nd April*, 80(26), 2012.

Michele Miranda, Elena Sofia Ruzzetti, Andrea Santilli, Fabio Massimo Zanzotto, Sébastien Bratières, and Emanuele Rodolà. Preserving privacy in large language models: A survey on current threats and solutions. *arXiv preprint arXiv:2408.05212*, 2024.

Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 78, 2004.

Zhuoyi Peng, Yixuan Tang, and Yi Yang. Adversarial mixup unlearning. *arXiv preprint arXiv:2502.10288*, 2025.

Formerly Data Protection. General data protection regulation (gdpr). *Intersoft Consulting, Accessed in October*, 24(1), 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Salah Rifai, Xavier Glorot, Yoshua Bengio, and Pascal Vincent. Adding noise to the input of a model trained with a regularized objective. *arXiv preprint arXiv:1104.3250*, 2011.

Jonas B Sandbrink. Artificial intelligence and biological misuse: Differentiating risks of language models and biological design tools. *arXiv preprint arXiv:2306.13952*, 2023.

Stefan Schoepf, Jack Foster, and Alexandra Brintrup. Potion: Towards poison unlearning. *arXiv preprint arXiv:2406.09173*, 2024.

Shaofei Shen, Chenhao Zhang, Yawen Zhao, Weitong Chen, Alina Bialkowski, and Miao Xu. Label-agnostic forgetting: a supervision-free unlearning in deep models. In *12th International Conference on Learning Representations, ICLR 2024*. International Conference on Learning Representations, ICLR, 2024.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. *arXiv preprint arXiv:2407.06460*, 2024.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

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

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pp. 303–319. IEEE, 2022.

Pengfei Wang, Zongzheng Wei, Heng Qi, Shaohua Wan, Yunming Xiao, Geng Sun, and Qiang Zhang. Mitigating poor data quality impact with federated unlearning for human-centric metaverse. *IEEE Journal on Selected Areas in Communications*, 42(4):832–849, 2023.

Weiqi Wang, Zhiyi Tian, Chenhan Zhang, and Shui Yu. Machine unlearning: A comprehensive survey. *arXiv preprint arXiv:2405.07406*, 2024a.

Weiqi Wang, Zhiyi Tian, Chenhan Zhang, and Shui Yu. Oblivious unlearning by learning: Machine unlearning without exposing erased data, 2025a. URL `https://openreview.net/forum?id=wAemQcyWqq`.

Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Parag Shah, Yujia Bao, Yang Liu, and Wei Wei. Llm unlearning via loss adjustment with only forget data. *arXiv preprint arXiv:2410.11143*, 2024b.

Yue Wang, Qizhou Wang, Feng Liu, Wei Huang, Yali Du, Xiaojiang Du, and Bo Han. GRU: Mitigating the Trade-off between Unlearning and Retention for Large Language Models. *arXiv preprint arXiv:2503.09117*, 2025b.

Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 191–195, 2021.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *arXiv preprint arXiv:2310.10683*, 2023.

Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, and Heng Ji. Unlearning Bias in Language Models by Partitioning Gradients. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 6032–6048, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.375. URL https://aclanthology.org/2023.findings-acl.375.

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

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

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. Does your LLM truly unlearn? An embarrassingly simple approach to recover unlearned knowledge. *arXiv preprint arXiv:2410.16454*, 2024c.

Chenxu Zhao, Wei Qian, Yangyi Li, Aobo Chen, and Mengdi Huai. Rethinking Adversarial Robustness in the Context of the Right to be Forgotten. In *International Conference on Machine Learning*, pp. 60927–60939. PMLR, 2024a.

Kairan Zhao, Meghdad Kurmanji, George-Octavian Bărbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. *Advances in Neural Information Processing Systems*, 37:12293–12333, 2024b.

Juexiao Zhou, Haoyang Li, Xingyu Liao, Bin Zhang, Wenjia He, Zhongxiao Li, Longxi Zhou, and Xin Gao. A unified method to revoke the private data of patients in intelligent healthcare with audit to forget. *Nature Communications*, 14(1):6255, 2023. Publisher: Nature Publishing Group UK London.

A Algorithm

Algorithm 1 Ready2Unlearn: Learning with Unlearning Preparedness

1: **Input:** Initial model parameters θ_0; training data $\mathcal{D}$, composed of forget data $\mathcal{D}_{\mathrm{f}}$ and retain data $\mathcal{D}_{\mathrm{r}}$; recovery data $\mathcal{D}_{\mathrm{rc}}$; number of outer-loop optimization steps N; adaptation rate α;[6] learning rate η; loss weight coefficients λ_1, λ_2, and λ_3; and loss function $\mathcal{L}$.
2: **Output:** A model with unlearning preparedness θ_P.
3: **for** $i = 1$ **to** N **do**
4: Sample $x_{\mathrm{f}} \sim \mathcal{D}_{\mathrm{f}}$, $x_{\mathrm{r}} \sim \mathcal{D}_{\mathrm{r}}$, $x_{\mathrm{rc}} \sim \mathcal{D}_{\mathrm{rc}}$
5: $\hat{\theta}_{i-1} = \theta_{i-1} + \alpha \nabla_{\theta_{i-1}} \mathcal{L}\left(\theta_{i-1}; x_{\mathrm{f}}\right)$ *# Inner-loop update mimicking the unlearning.*
6: $g_0 = \nabla_{\hat{\theta}_{i-1}} \mathcal{L}\left(\hat{\theta}_{i-1}; x_{\mathrm{f}}\right)$ *# For improving future unlearning efficiency.*
7: $g_1 = \nabla_{\hat{\theta}_{i-1}} \mathcal{L}\left(\hat{\theta}_{i-1}; x_{\mathrm{r}}\right)$ *# To support capability retention after future unlearning.*
8: $g_2 = \nabla_{\hat{\theta}_{i-1}} \mathcal{L}\left(\hat{\theta}_{i-1}; x_{\mathrm{rc}}\right)$ *# To enhance future post-unlearning resilience.*
9: Sample $x \sim \mathcal{D}$
10: $g_3 = \nabla_{\theta_{i-1}} \mathcal{L}\left(\theta_{i-1}; x\right)$ *# For maximizing current model utility.*
11: Update $\theta_i \leftarrow \theta_{i-1} - \eta\left(-g_0 + \lambda_1 g_1 + \lambda_2 g_2 + \lambda_3 g_3\right)$ *# Outer-loop parameter update.*
12: **end for**
13: $\theta_P \leftarrow \theta_N$
14: **return** θ_P

[6]Although the term "adaptation rate" does not refer to adaptation per se in our setting, we stick to it to maintain consistency with the meta-learning convention.

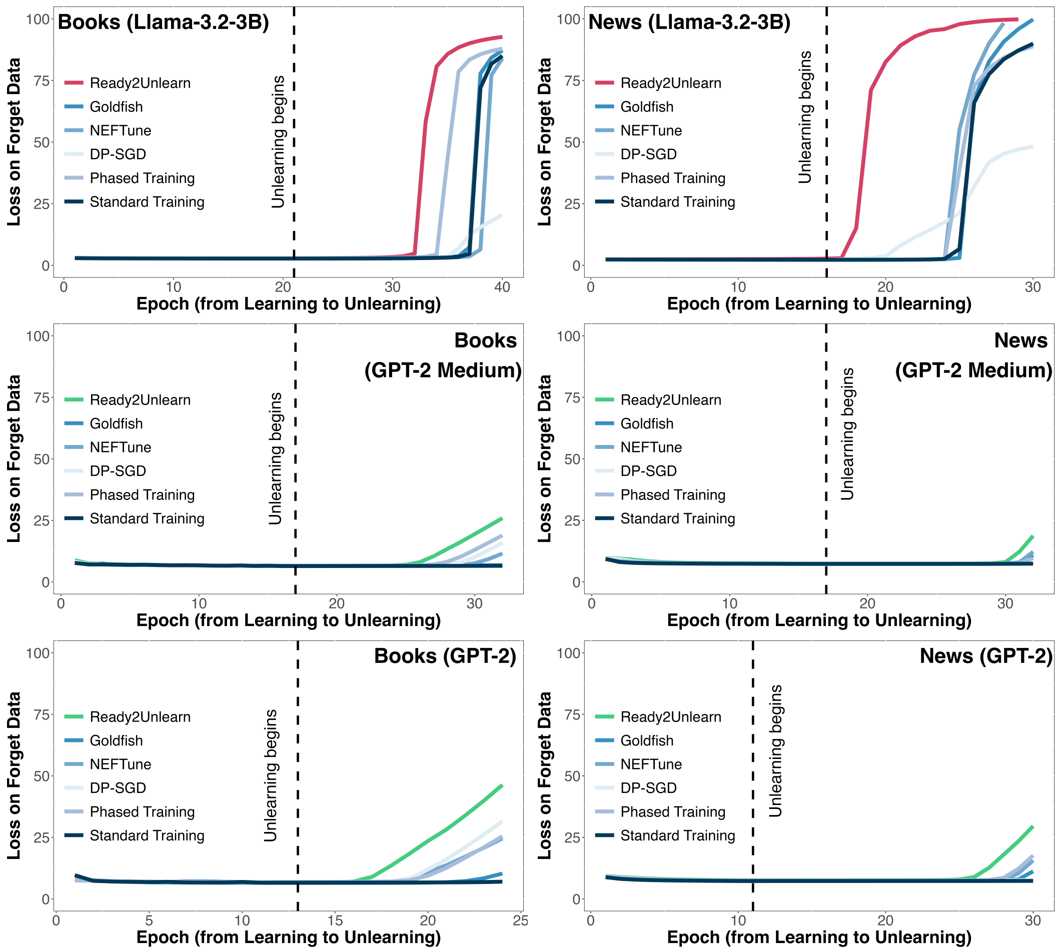

Figure 8: Extended comparison of unlearning efficiency in different language models: Llama-3.2-3B (top row), GPT-2 Medium (middle row), and GPT-2 (bottom row), on the MUSE-Books (left) and MUSE-News (right) datasets. The vertical axis represents the cross-entropy loss on the forget data, with each line corresponding to a different training-time strategy. During unlearning, we maintain a consistent unlearning rate of 1×10^{-6} across all methods to ensure a fair comparison. The vertical dashed line marks the moment when unlearning begins.

B Additional Unlearning Efficiency Evaluation in More Language Models

Please refer to Figure 8 for the evaluation results.

C Impact of Preparation Duration on Future Unlearning Efficiency

We investigate how the duration of preparatory training with Ready2Unlearn influences the efficiency of subsequent unlearning. Specifically, we explore how varying the number of epochs dedicated to preparation during training affects the model's ability to forget target data efficiently when unlearning is later initiated. We consider a total training budget of 20 epochs. For each experimental setting, we allocate the final M epochs to preparatory training using Ready2Unlearn, while the preceding $(20 - M)$ epochs follow Standard Training procedures. For example, the setting labeled "6 epochs" represents a scenario in which the model undergoes 14 epochs of Standard Training, followed by 6 epochs of preparation using Ready2Unlearn. The "20 epochs" setting then corre-

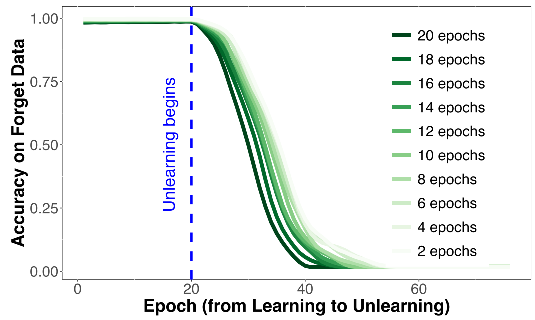

Figure 9: Effect of preparation duration on future unlearning efficiency with Ready2Unlearn. This figure visualizes the results for 10 distinct settings, where the number of epochs dedicated to preparation ranges from 2 to 20. The horizontal axis represents the timeline from the start of training to the completion of unlearning. The vertical axis shows the accuracy on the forget set, with lower values indicating more effective forgetting. The dashed line marks the epoch at which unlearning begins (epoch 20). All results are based on experiments conducted using the MNIST dataset.

sponds to applying Ready2Unlearn throughout the entire training period. The results for 10 distinct settings ($M \in \{2, 4, 6, \ldots, 20\}$), presented in Figure 9, reveal a clear trend that longer preparation with Ready2Unlearn consistently leads to a faster response to future unlearning requests, with the model's performance on the forget data dropping earlier. This suggests that more extensive integration of Ready2Unlearn during training equips the model with stronger unlearning readiness.