In-Context Unlearning: Language Models as Few-Shot Unlearners

Martin Pawelczyk [1] **Seth Neel** [* 1] **Himabindu Lakkaraju** [* 1]

Abstract

Machine unlearning, the study of efficiently removing the impact of specific training instances on a model, has garnered increased attention in recent years due to regulatory guidelines such as the *Right to be Forgotten*. Achieving precise unlearning typically involves fully retraining the model and is computationally infeasible in case of very large models such as Large Language Models (LLMs). To this end, recent work has proposed several algorithms which approximate the removal of training data without retraining the model. These algorithms crucially rely on access to the model parameters in order to update them, an assumption that may not hold in practice due to computational constraints or having only query access to the LLMs. In this work, we propose a new class of unlearning methods for LLMs called "In-Context Unlearning." This method unlearns instances from the model by simply providing specific kinds of inputs in context, without the need to update model parameters. To unlearn specific training instances, we present these instances to the LLMs at inference time along with labels that differ from their ground truth. Our experimental results demonstrate that in-context unlearning performs on par with, or in some cases outperforms other state-of-the-art methods that require access to model parameters, effectively removing the influence of specific instances on the model while preserving test accuracy.

1. Introduction

Over the past decade, machine learning (ML) models have become ubiquitous in high-stakes decision making settings such as hiring, criminal justice, and credit scoring. To ensure responsible deployment and usage of these models in real-world applications, several regulatory guidelines have been introduced to protect user privacy (OAG, 2021; Union, 2016), one of which is called the *Right to be Forgotten*. The Right to be Forgotten offers users more control over their personal data by allowing them to submit a deletion request to retract permission for a company to use their personal data at any time, even if, for example, the company has already trained an ML model with it (Biega et al., 2020; Goldsteen et al., 2021; OAG, 2021; Union, 2016). This raises a significant dilemma for organizations that aim to comply with the spirit of the regulation and avoid potentially breaking the law (Voigt & Von dem Bussche, 2017), particularly concerning how exactly this data should be "removed" from any models trained on it. A second motivation comes from a concern orthogonal to privacy: copyright infringement. Generative models, like Large Language Models (LLMs), can often reproduce their training data verbatim or with only superficial changes. This can lead to credible claims of copyright infringement when the underlying data is protected by copyright. In such cases, when a copyrighted input to a model is generated, the model owner may be required to "take down" the copyrighted work from the model. Unlearning the corresponding input is one potential technical solution to comply with the take down request without retraining the model from scratch.

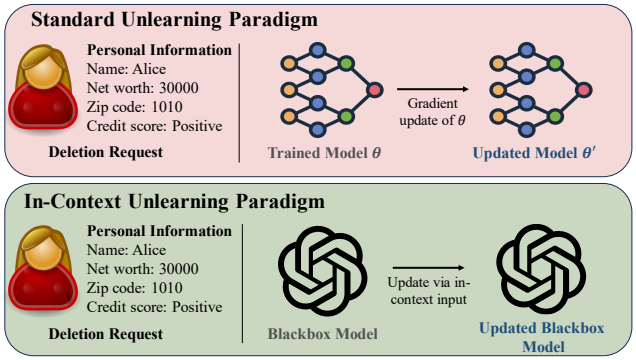


Figure 1. **Differences between In-Context Unlearning and Standard Unlearning**. **Top**: Traditional unlearning approaches require access to model parameters θ and these parameters are updated in response to deletion requests. **Bottom**: In-context unlearning does not require access to the parameters. Unlearning works by providing certain kinds of inputs in context which mimic the model's performance as if the model was re-trained without the points.

[*]Equal contribution [1]Harvard University, US. Correspondence to: Martin Pawelczyk <martin.pawelczyk.1@gmail.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

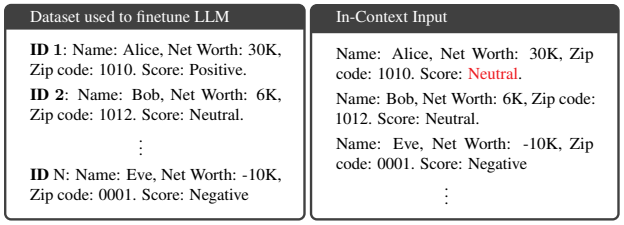


Figure 2. **Demonstrating in-context unlearning**. **Left**: The data set used to finetune the LLM. **Right**: In-context unlearning removes the influence that samples from the forget set S_f (**ID 1** from the dataset) have on the completion by adding examples from the forget set with different labels to the in-context input (e.g., for "Name: Alice, Net Worth: 300K, Zip code: 1010" the label was changed randomly from Positive to Neutral).

Indeed as a recent paper by Henderson et al. (2023) on copyright issues in generative models remarks: *retraining a model without a taken down datapoint could be exceedingly costly ... new research is needed to identify new and improved mechanisms for handling takedown requests in this relatively new setting.* Work in "Machine Unlearning" seeks to bridge this gap by building algorithms that remove the influence of the deleted point from a trained model, while avoiding the computationally expensive step of fully re-training on the updated dataset (Ginart et al., 2019; Sekhari et al., 2021).

At the same time as ML privacy regulation has started to gain traction, the release of Large Language Models (LLMs) has marked a pivotal transition in ML research (Brown et al., 2020). Modern LLMs have demonstrated competency in a vast array of challenging tasks, ranging from language comprehension (Radford et al., 2019), reasoning (Bubeck et al., 2023) to tabular data generation (Borisov et al., 2023). These models not only exhibit effective abilities on tasks they were designed for, but they also display remarkable adaptability to unfamiliar tasks. This surprising versatility is partially attributed to a learning paradigm called "in-context learning" (Brown et al., 2020), wherein the model has access to a set of in-context examples, a minimal collection of input and label pairs, that are added to the prompt at inference time to enhance LLM performance.

Despite the prominence of LLMs, and extensive recent work on machine unlearning, studying unlearning in LLMs is relatively unexplored (Jang et al., 2023). Perhaps this is because compared to conventional machine unlearning on image classifiers for example, unlearning in LLMs has two additional challenges. First, many LLMs operate as black-boxes (see Figure 1), meaning that standard unlearning techniques that perform gradient ascent or descent on the model's parameters cannot be implemented (Neel et al., 2021). Second, even if the unlearning algorithm has "white-box" access (access to model parameters), performing gradient updates on LLMs with many billions of parameters every time an unlearning request comes in might be computationally infeasible.

To address these challenges, we propose a novel class of unlearning methods suitable for large language models (see Figure 1). To the best of our knowledge, this work is the first to suggest In-Context UnLearning (`ICUL`) which deploys a uniquely built context to eliminate the influence of a training point on the model output. In order to unlearn a particular training instance, the model context is constructed in such a way that the labels associated with training points targeted for deletion are flipped randomly, and both the training points and their flipped labels are provided at the beginning of the context alongside additional correctly labelled context examples sampled from the training data distribution (see Figure 2). Our `ICUL` method does not require knowledge of the LLM's parameters, and yet manages to achieve performance levels that are competitive with or in some cases exceed the state-of-the-art LLM unlearning methods which require access to LLM parameters and involve expensive gradient computations (Jang et al., 2023).

We experiment with multiple established real world datasets: AG-News, SST-2, SQUAD, and Amazon reviews to evaluate the effectiveness of our proposed unlearning method. Our results on text classification and question-answering tasks clearly demonstrate the efficacy of the proposed unlearning method, and highlight that it practically eliminates a training point's influence on the model output. These results indicate the significant potential for unlearning training points from black-box models. Our proposed methods and findings offer a new perspective on unlearning mechanisms in LLMs:

- **New unlearning paradigm for LLMs**: This is the first work to use in-context learning for machine unlearning by specifically constructing contexts that induce model behavior that is indistinguishable from the behavior of a re-trained model.
- **New empirical unlearning evaluation:** In Subsection 3.2 we introduce `LiRA-Forget`, a new unlearning evaluation that adapts the LiRA MIA (Carlini et al., 2022) to the problem of evaluating unlearning. We note that a similar evaluation metric was introduced concurrently by Kurmanji et al. (2023), and subsequently studied by Hayes et al. (2024) under the name U-LiRA. We discuss these works more in Section 2.
- **Data deletion from blackbox models**: `ICUL` does not require access to model parameters and can be readily applied to blackbox models. This makes it a useful tool to patch a model until the model can be updated or a retrained version can be deployed at the next deployment phase. Thus, it is complementary to existing white-box unlearning techniques which have higher computational burdens.
- **Lower memory requirements**: Our method boasts

lower memory requirements compared to state-of-the-art unlearning methods like Gradient Ascent (GA), especially as the size of the LLM increases. For instance, on Llama-2 7B, ICUL runs on a Tesla V100 GPU with 32GB of RAM, while running GA would require access to an A100 GPU with 80GB of RAM. This makes ICUL computationally feasible for LLMs with billions of parameters.

2. Related Work

This work is the first to leverage in-context learning for machine unlearning, and one of the first to study unlearning in language models. Below we discuss related works for each of these topics.

In-Context Learning. Transformers form the foundation of contemporary LLM architectures. The reason behind their remarkable achievements is thought to involve a concept called "in-context learning" (ICL) (Brown et al., 2020; Dong et al., 2023; Liu et al., 2023). This refers to their ability to adapt to new tasks flexibly by incorporating data provided in the context of the input sequence itself, rather than fine-tuning which explicitly updates weights. Exploring the full capabilities of ICL remains an active area of research, with recent works trying to understand its potential better empirically by studying in-context example design (Garg et al., 2022; Liu et al., 2022; 2023; Min et al., 2022). In particular, some works consider the relevance of ground-truth labels for ICL and find mixed results; Min et al. (2022) find that ground-truth labels have little impact on classification performance while the findings by Wei et al. (2023) suggest that only larger scale LLMs can adopt their predictions to align with flipped label contexts. While all these works study how learning can be facilitated through in-context examples, none of these works explore how unlearning can be achieved by designing in-context examples.

Machine Unlearning. Motivated by GDPR's "Right to be Forgotten" recent literature develops procedures for updating machine learning models to remove the impact of training on a subset of points (Ginart et al., 2019; Golatkar et al., 2020a;b; Huang & Canonne, 2023; Izzo et al., 2021; Jang et al., 2023; Neel et al., 2021; Sekhari et al., 2021; Wang et al., 2023; Wu et al., 2020) or a subset of concepts (Belrose et al., 2023; Ravfogel et al., 2022a;b) without having to retrain the entire model from scratch. Unlearning algorithms fall into two camps: *exact unlearning* approaches that redesign training in order to permit efficient re-training (e.g., Ginart et al. (2019); Sekhari et al. (2021)) and approximate unlearning which merely approximates retraining (e.g., Jang et al. (2023); Neel et al. (2021)). The latter approach has been likened to "forgetting" (Graves et al., 2021; Jagielski et al., 2023; Tirumala et al., 2022) which tracks whether machine learning models progressively unlearn samples during the course of training and is typically quantitatively assessed by membership inference (MI) attack accuracy (Jagielski et al., 2023). For simple hypothesis classes such as linear regression (Cook & Weisberg, 1980; Guo et al., 2019; Izzo et al., 2021) or kernel methods (Pawelczyk et al., 2023; Zhang & Zhang, 2021), tailored machine unlearning methods exist that make use of closed for solutions for the updated model.

For the majority of non-convex models used in practice, in order to preserve accuracy the training routine is left untouched. As a result, standard approximate unlearning algorithms do not have theoretical guarantees, and so they must be evaluated empirically. Prior to this work, these empirical guarantees typically relied on heuristics like comparing the loss of unlearned points to the average validation loss (Jang et al., 2023), or aggregate properties of the unlearned model like the test error, error on the forget set, or distribution of model confidences on unlearned and test points (Golatkar et al., 2020b). Our work proposes a more principled empirical unlearning evaluation based on constructing an optimal membership inference attack (MIA) to distinguish unlearned points from test points, based on the LiRA MIA (Carlini et al., 2022), which is the state of the art MIA. We note that Kurmanji et al. (2023) concurrently propose a similar evaluation they call LiRA-for-unlearning, and subsequently Hayes et al. (2024) perform a detailed benchmarking of existing unlearning algorithms using this metric, as well as more heuristic unlearning metrics. Critically they find that evaluations that do not evaluate the unlearning of a specific point $\mathbf{x}$ using an example-specific threshold, tend to over-estimate unlearning performance relative to LiRA-based techniques.

Prior research has mostly explored unlearning from discriminative classifiers, generally vision models (e.g., Goel et al. (2022); Golatkar et al. (2020a)), where the aim often is to forget entire classes like "cats" or "ships." These approaches typically update the model by starting at the model produced after training, and taking either gradient ascent steps on the deleted points (Jang et al., 2023) or gradient descent steps on the retained points (Neel et al., 2021), sometimes combining both approaches simultaneously and adding regularization (Foster et al., 2023; Jia et al., 2024; Kurmanji et al., 2023).

3. Preliminaries

Here, we first discuss the generic formulations of in-context learning. We then discuss how to measure unlearning success empirically.

3.1. In-Context Learning

In-context learning has recently emerged as a new paradigm that allows auto-regressive language models to learn tasks

using a few examples in the form of context demonstrations (Brown et al., 2020). Here, we follow common practice (Brown et al., 2020; Dong et al., 2023; Liu et al., 2023), and consider the following definition of in-context learning: For a given pretrained language model f_θ, a set of context demonstrations D_{context} and a query input, the language model generates a sequence of tokens with a predefined length. For example, when the model is used for text classification, it typically outputs one additional token as its prediction from a set of C possible tokens where C is usually large (e.g., for the Bloom model $C = 250680$). The context D_{context} consists of an optional task instruction and L demonstration examples; D_{context} = {[Instruction input] [Example input 1] [Label 1], ... [Example input L] [Label L]}. The prompt, which uses D_{context} along with the query [Query Input], is then provided as input for the language model prediction. In-context learning has emerged as a way to improve a pretrained model's predictions without the need of costly finetuning of the model for a specific task.

3.2. `LiRA-Forget`: Measuring Unlearning

We now define how we measure (approximate) unlearning. Our unlearning notion is that of Ginart et al. (2019); Neel et al. (2021), but adapts the metric of MI attack success to operationalize this definition (Goel et al., 2022; Golatkar et al., 2021). Let $S \subset \mathcal{S}^*$ denote the training set, sampled from a distribution $\mathcal{D}$. Let $\mathcal{T} : \mathcal{S}^* \to \Theta$ be the (randomized) training algorithm that maps S to a parameterized model $f_{\theta(S)}$. Further define the forget set as the subset of points to be forgotten from the trained machine learning model denoted by $S_f \subset S$. We define an unlearning procedure $\mathcal{U}$ that takes as input the model $f_{\theta(S)}$, the forget set S_f of data samples that should be deleted, and the train set S (and possibly some auxiliary information which we suppress), and outputs an updated model $\bar{f} \sim \mathcal{U}(f_{\theta(S)}, S, S_f)$. Denote the probability law of the training algorithm on input S by p_S, the law of the exact re-training algorithm by $p_{S\setminus S_f}$, and the law of the unlearning algorithm by $p_{\mathcal{U}}$. As first formalized in (Ginart et al., 2019), the goal of an approximate unlearning algorithm is to achieve small $d(p_{S\setminus S_f}, p_{\mathcal{U}})$ for some distance measure between distributions d. Empirically verifying whether $d(p_{S\setminus S_f}, p_{\mathcal{U}})$ is small is difficult for two reasons: i) For computational reasons we do not have direct access to samples from $p_{S\setminus S_f}$, and ii) even if we did these distributions are extremely high dimensional and cannot be compared efficiently.

We address issue (i) by approximating the re-training distribution via sample-splitting (described in more detail in Appendix B); by training multiple models on splits of the data that do not contain S_f, we can approximate samples from $p_{S\setminus S_f}$. This approach is known as training "shadow-models" and has been employed for MI in Shokri et al. (2017). We address (ii) by re-formulating the problem of bounding $d(p_{\mathcal{U}}, p_{S\setminus S_f})$ as a hypothesis testing problem. Le Cam's Lemma (see Theorem 2.2 in Tsybakov (2008)) establishes a correspondence between $d(p_{\mathcal{U}}, p_{S\setminus S_f})$ and the ability of an optimal hypothesis test to distinguish $p_{\mathcal{U}}$ from $p_{S\setminus S_f}$ based on a single sample. More specifically, we imagine a model f is sampled from $p_{\mathcal{U}}$ with probability $1/2$ else from $p_{S\setminus S_f}$ with probability $1/2$, and conduct a hypothesis test to determine which distribution f came from:

$$\mathrm{H}_0 : f \sim p_{S\setminus S_f} \text{ vs. } \mathrm{H}_1 : f \sim p_{\mathcal{U}}. \tag{1}$$

Rejecting the null hypothesis corresponds to inferring that f was not from the re-training distribution. The Neyman-Pearson lemma (Neyman & Pearson, 1933) asserts that the optimal hypothesis test at a predetermined false-positive rate involves thresholding the likelihood-ratio test statistic Λ. As discussed, approximating the exact likelihood ratio statistic Λ is intractable due to the high dimensionality of f, and so we follow recent work on MIAs, that instead takes the likelihood ratio with respect to the distribution of losses on the forget points S_f for both models. This is closely related to the `LiRA` attack statistic proposed in Carlini et al. (2022), but differs critically in that the numerator considers the model produced by training on S_f *and then unlearning* via $\mathcal{U}$ rather than the model that results after training. We then define the `LiRA-Forget` statistic $\hat{\Lambda}$:

$$\hat{\Lambda} = \frac{\prod_{(\mathbf{x},\mathbf{y})\in S_f} p_{\mathcal{U}}\big(\ell\big(f(\mathbf{x}),\mathbf{y}\big)\big)}{\prod_{(\mathbf{x},\mathbf{y})\in S_f} p_{S\setminus S_f}\big(\ell\big(f(\mathbf{x}),\mathbf{y}\big)\big)}, \tag{2}$$

where ℓ denotes an appropriate loss function. As in these recent works we approximate the univariate distributions on losses in the numerator and denominator of (2) via sample-splitting. Specifically we fine-tune models on sub-sampled datasetes that either contain or do not contain S_f. To approximate the numerator, on the datasets that do contain S_f, we run $\mathcal{U}$ to unlearn S_f, and then compute the updated model's loss on S_f. To approximate the denominator, we simply take the models that were not trained on S_f and compute their losses on S_f.

Operationalizing `LiRA-Forget`. Operationalizing the likelihood ratio test from (2) requires access to the distribution of losses under the null and alternative hypotheses. While analytical solutions are usually not available, we can readily get large samples from these two distributions. In an ideal scenario, this entails that we would need to fit as many re-train models and unlearned models as possible for every forget set of interest. Since this approach becomes computationally too burdensome, we use the following approximation. We adapt the sample splitting procedure first introduced by Carlini et al. (2022) to forget sets with sizes $J = \{1, 5, 10, 20\}$, and approximate the distributions under H_0 and H_1 from equation (2). We train K shadow models

on random samples from the data distribution $\mathcal{D}$ so that a fraction p of these models are trained on the forget set $S_f = \{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1}^{J}$, and a fraction $(1-p)$ are not. In particular, we train shadow models on $K = 10$ subsets of $\mathcal{D}$ so that each forget set $S_f \in S$ appears in $K \cdot p$ subsets. This approach has the advantage that the same K shadow models can be used to estimate the likelihood-ratio test for all the forget sets. Finally, we fit the parameters of two Gaussian distributions to the confidence scores of the retain models and the unlearned models on S_f.

4. Our Framework: In-Context Unlearning

In this section, we describe our framework, In-Context Unlearning (`ICUL`), in detail. Recall that the main goal of our framework is to eliminate the need to re-train the model from scratch or to update the parameters of the model when unlearning specific training data points. Instead, at inference time, we construct a specific context which lets a language model trained on a classification or question-answering task a behave as if it had never seen the specific data point during training before.

To this end, our framework leverages both correctly labeled / answered as well as mislabeled / incorrectly answered examples to construct an in-context input which is provided as input to the LLM at inference time. By changing the labels / answers on the points targeted for unlearning, our method diminishes the model's confidence specifically on these instances, aligning them more closely with the model's confidence in the case where the points were never part of the training set. In particular, the label / answer changing operation in Step (1) below aims to remove the influence a specific training point has on the model outcome. Since Step (1) may cause the model to "overcorrect" on the forget points leading to decreased test accuracy and invalid unlearning, Step (2) from below serves as an efficient way to dampen the effect of flipping the labels / answers of forget points. More specifically, we suggest the following 3 step in-context input construction approach which we term `ICUL` and which we illustrate for a classification task:

1) **Change labels on forget points to different labels.** Given a deletion request of size K, we randomly flip the labels on the corresponding K training points whose influence should be removed from the model resulting in the template: "[*Forget Input 1*] [*Different Label*] $\cdots$ [*Forget Input K*] [*Different Label*]".
2) **Add L correctly labeled training points.** We randomly sample L labeled examples and add them to the template of step 1, resulting in the updated template: "[*Forget Input 1*] [*Different Label*] $\cdots$ [*Forget Input K*] [*Different Label*] $\backslash n$ [*Input 1*] [*Label 1*] $\cdots$ [*Input L*] [*Label L*]".
3) **Prediction.** Finally, we add the query input to the template resulting in the final prompt "[*Forget Input 1*] [*Different Label*] $\cdots$ [*Forget Input K*] [*Different Label*] $\backslash n$ [*Input 1*][*Label 1*] $\cdots$ [*Input L*] [*Label L*] [*Query Input*]" and let the model predict the next token using temperature $t = 0$.

If the underlying task is question-answering, then we analogously design the contexts by flipping the answers for the forget-targeted samples to other random answers from the dataset.

5. Empirical Evaluation

We now present our empirical analysis. First, we empirically show that in-context unlearning is successful at unlearning information from a finetuned LLM in a forward pass. In Section 5.2, we show that the unlearned model maintains extremely competitive model performance when using `ICUL` especially with larger deletion requests of 10 or 20 points despite having no access to model parameters. In Section 5.3 we demonstrate that our method also works reliably for larger LLMs like Llama-2 (7B), and finally in Section 5.5 we show a variety of ablation experiments that emphasize that our method works as intended. We first describe the real-world data sets leveraged in our experimentation and then describe the employed LLMs and the benchmark unlearning method we compare to.

Datasets. We evaluate our in-context input constructions on 3 standard text classification tasks, Stanford Sentiment Treebank (SST2) (Socher et al., 2013), Amazon polarity, and AG-News (Zhang et al., 2015). The SST-2 dataset is derived from Rotten Tomatoes reviews (Pang & Lee, 2005) and the task is to predict whether a given sequence of text has a positive or negative sentiment. We also use the Amazon polarity and the AG-News datasets which were originally introduced by Zhang et al. (2015). For Amazon, the task is binary classification for whether a given review is positive (four or five stars) or negative (one or two stars). For the AG-News dataset the task consists of classifying news articles in one of four classes 'sports', 'business', 'world' or 'technology'. We also provide experiments on the standard SQUAD dataset (Rajpurkar et al., 2016), which represents a question-answering task. In line with work on auditing privacy leakages (Carlini et al., 2023; Shokri et al., 2017), we randomly sub sampled smaller data sets of 25000 points from each of these datasets for finetuning. We show the average results over 10 runs for all of our experimental settings and usually report ± 1 standard deviation across these runs.

Large Language Models. We conduct experiments on Bloom (560M, 1.1B, 3B) (Scao et al., 2022) and Llama-2 (7B) (Touvron et al., 2023) LLMs. We finetune these models on the classification datasets using the following

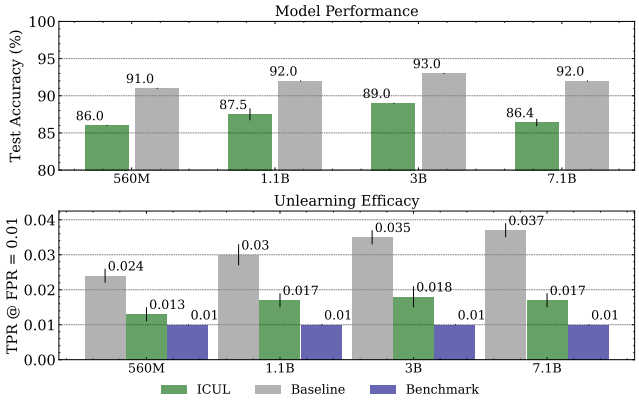


Figure 3. **Evaluating `ICUL` sensitivity across model sizes.** We empirically assess unlearning via `ICUL` with $L = 6$ for 10 deletion requests on Bloom LLMs (560M, 1.1B, 3B, 7.1B) finetuned on the SST-2 dataset. `Baseline` indicates performance when no unlearning is conducted, while `Benchmark` indicates best possible performance. Vertical bars show ± 1 standard deviation across 10 evaluation runs.

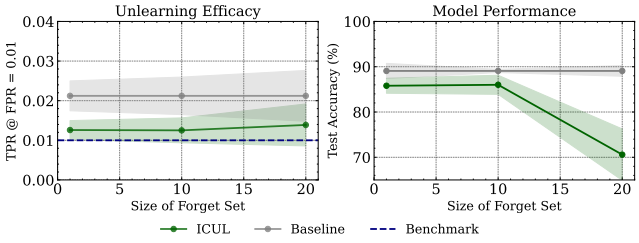


Figure 4. **`ICUL` is effective on state-of-the-art LLMs.** We conduct empirical evaluations on unlearning via `ICUL` with varying numbers of deletion requests $(1, 10, 20)$ for a Llama2 (7B) LLM fine-tuned on the SST-2 dataset. Finetuning required an A100 GPU (80 GB), while unlearning through `ICUL` with $L = 6$ was performed at inference time using a V100 GPU (32 GB). `Baseline` indicates performance when no unlearning is conducted, while `Benchmark` indicates best possible performance. Shades indicate ± 1 standard deviation across 10 evaluation runs.

template for each sample: *"[Input] [Label]"*. We use the standard causal cross-entropy loss with initial learning rate set to $5 \cdot 10^{-5}$ which encourages the model to predict the next token correctly given a total vocabulary of C possible tokens, where C is usually large (e.g., for the Bloom model $C = 250680$). At test time, the models predict the next token from their vocabularies given a context and query.

Prior Baselines. We implement the only available baseline for unlearning in LLMs from Jang et al. (2023); they unlearn via gradient ascent on the forget set, which can be interpreted as maximizing instead of minimizing the loss on the forget points. We follow their suggestion and set the learning rate to $5 \cdot 10^{-5}$, use one epoch and do sequential unlearning where every point from the forget set is individually and sequentially unlearned using a constant learning rate schedule. Additionally, since a learning rate of $5 \cdot 10^{-5}$ usually led to poor results, we did a search over different learning rates $\{5 \cdot 10^{-5}, 3 \cdot 10^{-5}, 1 \cdot 10^{-5}, 5 \cdot 10^{-6}\}$. In the main text, we report the most competitive results.

5.1. Evaluation Measures

When evaluating the efficacy of an unlearning method $\mathcal{U}$, three distinct objectives emerge: validity of the unlearning procedure, classification accuracy post-unlearning, and runtime and space requirements of the algorithm. We first discuss measures that gauge unlearning validity. We have described how to compute our unlearning success statistic $\hat{\Lambda}$, but it remains to discuss what values of $\hat{\Lambda}$ should be considered "successful". We continue our analogy to recent work in evaluating membership inference attacks, and follow the paradigm introduced in (Carlini et al., 2022; Leemann et al., 2023) that focuses on true positive rates (in this case of predicting that the loss came from the unlearned model) at low false positive rates as the most intuitive measure of MIA attack success. Unlike in the MIA context, where a successful attack has an AUC $\gg .5$, and an ROC curve that is above the diagonal even at very low FPRs, in our setting a successful unlearning algorithm corresponds to the failure of the LRT, and so we hope to see ROC curves that are very close to the diagonal even at low FPRs.

`Benchmark`: Random guessing performance. The first measure consists of the decision not to unlearn the point from the model. It is represented by the dotted diagonal line indicating an equal ratio of FPR to TPR denoted as `Benchmark`. For lower FPRs below 10^{-1}, an unlearning method should demonstrate performance as close to the random guessing `Benchmark` as possible and below the `Baseline`, which we discuss next.

`Baseline`: Train vs. held out samples on the initial model $f_{\theta(S)}$. This evaluation is a starting point measuring the initial information leakage from the model. It consists of the decision not to unlearn the point from the model and we will denote this as `Baseline` in all figures. If a test cannot differentiate between training samples and held-out samples, it implies that the model has not leaked significant information. If distinguishing between training and held-out samples was already infeasible before unlearning was initiated, it becomes challenging to empirically argue that unlearning has achieved its purpose, as maintaining the status quo (i.e., doing nothing) would be a reasonable strategy. To conduct this evaluation, we run the `LiRA` attack using 10 shadow models (Carlini et al., 2022) on the model $f_{\theta(S)}$.

Forget vs. held out samples on the updated model $\bar{f}$. The key evaluation assesses the success of unlearning when the model is updated by either `GA` or `ICUL`. Can the model effectively forget the specific data point in question? I.e, is the model output on a data point when it is held out of the train-

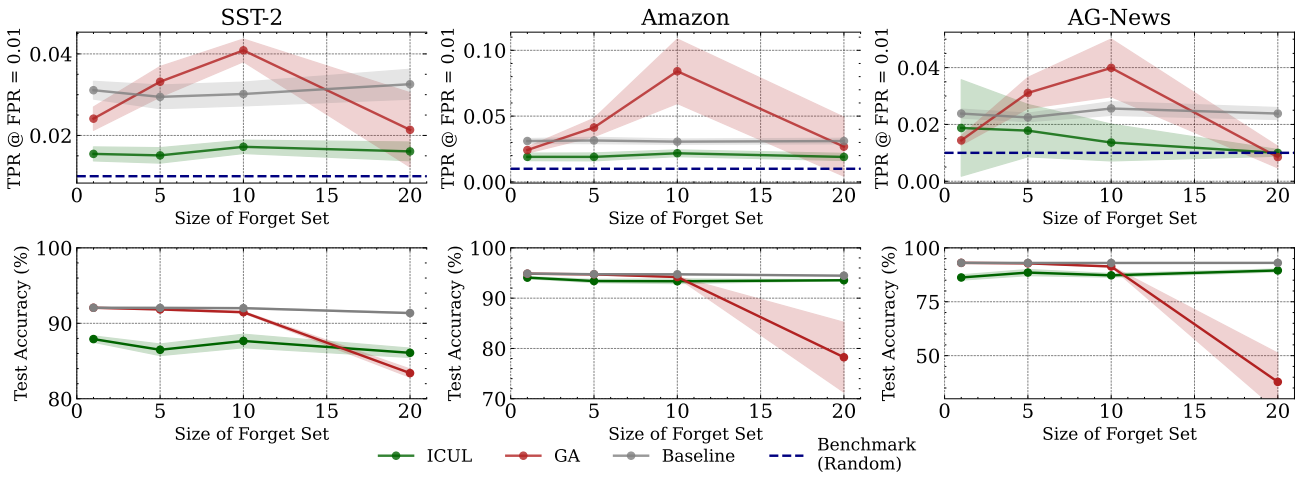


Figure 5. **Evaluating unlearning.** We empirically evaluate unlearning across different unlearning methods for different number of deletion requests (1, 5, 10, 20) for a Bloom 1.1B model finetuned on different data sets (columns). For ICUL, we select the most competitive results for $L \in \{2, 4, 6\}$, while for GA we search over learning rates of $\{5 \cdot 10^{-5}, 3 \cdot 10^{-5}, 1 \cdot 10^{-5}, 5 \cdot 10^{-6}\}$. Shades indicate ± 1 standard deviation across 10 evaluation runs. Reproducing these experiments requires approx. 1800 GPU hours on a V100 GPU (32GB). **Top row – Unlearning efficacy**: LiRA-Forget performance at a fixed FPR=0.01. Baseline indicates performance when no unlearning is conducted. Ideally, GA and ICUL performance curves trace significantly below the Baseline and as close to the random guessing Benchmark (dashed line) as possible. **Bottom row – Model Performance**: Test accuracies as we vary the number of deletion requests.

ing set indistinguishable from the output on the same data point when it was initially part of the model but subsequently removed through the unlearning process? This critical evaluation is conducted by running our LiRA-Forget attack against the model $\bar{f}$ as discussed in Section 3.

Evaluating model performance. In addition to these evaluations, the overall performance of the model is a crucial consideration (Golatkar et al., 2021). The model's predictive capabilities should demonstrate effectiveness across various scenarios, including 1) train points S, 2) points S_f targeted for unlearning and 3) randomly drawn test points.

5.2. Empirically Evaluating Unlearning

In this Section, we evaluate the efficacy of unlearning various numbers of deletion requests $(1, 5, 10, 20)$ for a Bloom 1.1B model for all considered data sets. The results are summarized in Figure 5. We compare GA, which has access to model parameters, with our proposed ICUL method, and compare their performance to the Baseline and the random guessing Benchmark.

Inspecting the top row of Figure 5, we find the ICUL curves, for all datasets and sizes of deletion requests, trace close to the Benchmark that represents a random guess probability of whether a point intended for removal is still part of the LLM. Also note that our method consistently surpasses the Baseline in terms of TPRs at FPRs of 0.01 on all datasets and across all deletion requests. When we contrast ICUL with GA, ICUL consistently achieves superior (lower) TPRs at FPRs of 0.01, besting GA in 14 out of 16 cases. Surprisingly, these results show conclusively that ICUL unlearns more effectively than GA using the strong LiRA-Forget evaluation, despite being moch more memory efficient (see App. C), and requiring only black-box model access.

Figure 5 also reveals intriguing patterns regarding GA: for both small and large deletion requests, GA demonstrates reasonable performance, outperforming Baseline, but in the middle ground of 5 and 10 deletion requests performance drops below even the trivial baseline.

Evaluating the unlearned models' performance. Next, we assess the performance of the models post-unlearning, using accuracy as the evaluation metric. An overview of these results can be found in the bottom row of Figure 5. For results on the forget points' and train points' performance see Figure 7 in the Appendix. In the main text, we focus on performance on the test points. While GA exhibits better test accuracy than ICUL, as we expand the number of deletion requests to 10, the performance gap between ICUL and GA on unseen test data starts narrowing down. Remarkably, for 20 deletion requests, the performance of GA drops significantly while ICUL maintains a similar level of test accuracy regardless of the number of deletion requests. Even below deletion requests of size 20, ICUL obtains reasonable test accuracy on all datasets, and on Amazon the test accuracy is within 1% of the performance of the Baseline.

5.3. Sensitivity of In-Context Unlearning to Model Size

Impact of varying model size on ICUL. When assessing ICUL across varying model sizes, two discernible trends

	Accuracy	TPR @ FPR = 10^{-2}	Benchmark
5 Deletions			
Baseline	72.0%	0.0241	0.01
ICUL	68.7%	0.0183	0.01
10 Deletions			
Baseline	72.2%	0.0247	0.01
ICUL	60.3%	0.0140	0.01

Table 1. Comparison of performance metrics for Baseline and ICUL methods with 5 and 10 deletions when performing unlearning on the SQUAD dataset.

emerge (refer to Figure 3). First, somewhat surprisingly, there is a positive correlation between model size and the percentage improvement over the Baseline: 41.67% for the 560M LLM, 43.33% for the 1.1B LLM, for the 3B LLM 51.42%, and for the 7.1B LLM 54.05% (see Figure 3, bottom). Second, we observe a relationship between the number of parameters in the LLM and the post-unlearning model performance, as evidenced by an increase in test accuracy from 86.1% for the 560M LLM to 89% for the 3B LLM (see Figure 3, top).

Exploring sensitivity across state-of-the-art LLMs. Here, we examine the sensitivity of ICUL results to the class of language models employed. To gauge this, we assess the performance of our unlearning algorithm in erasing 1, 10, and 20 points from the SST-2 dataset using the Llama-2 (7B) LLM, recognized as one of the top-tier models within the 7B parameter class of LLMs (Touvron et al., 2023). The findings, depicted in Figure 4, affirm that ICUL extends its effectiveness to other classes of LLMs, exhibiting forgetting efficacy comparable to that observed in Bloom models. Notably, akin to the smaller Bloom models, the TPR at a FPR of 0.01 closely aligns with the Benchmark across all deletion request sizes.

5.4. Broadening the Scope of In-Context Unlearning

We explore the possibility of using ICUL on additional language tasks, conducting empirical evaluations on a question answering task using the SQUAD dataset (Rajpurkar et al., 2016). We focus on unlearning 5 and 10 samples from the finetuned model. The results summarized in Table 1 demonstrate that ICUL is effective beyond multiclass classification tasks; ICUL reduces privacy leakage by 24% for 5 deletion requests and by 43% for 10 deletion requests compared to the Baseline, performing close to the Benchmark. While the accuracy drop is a moderate 5.5% for 5 deletions, it becomes a significant 16.7% for 10 deletions.

5.5. Towards Understanding In-Context Unlearning

Next, we study the factors in the context construction that lead to successful in-context unlearning, conducting additional analyses where we vary the context length, label flipping, and role of the forget point in the ICUL context from Steps 1 and 2.

Varying context length. One key factor to consider is the length of the context. This might influence the unlearning process. So, our framework considers a few different context lengths by varying the total number of correctly labelled context examples $L \in \{2, 4, 6\}$, which we refer to as ICUL(L). For ICUL, changing the context length can significantly improve results as seen on the right in Figure 6. With shorter context lengths, such as 2, the reversed label of the forget point typically leaves an overly negative impact on the model's confidence scores. This generally results in poorer average performance than the Baseline, as shown by the comparison of their AUC scores (e.g., ICUL(2) scores at 0.67 while Baseline at 0.58). Furthermore, context lengths of this size are often not sufficient enough to reduce TPRs at FPR levels of $\{10^{-3}, 10^{-2}, 10^{-1}\}$ down to the level of random guessing benchmark. On the other hand, 4 or 6 additional context examples tend to yield the best performance.

ICL. Another crucial consideration is examining the necessity of label flipping for successful unlearning, and including a baseline where we avoid label flipping of the point that should be unlearned from step 1, which results in the following in-context input: *"*[*Forget Input*] [*Label*] \n [*Input 1*] [*Label 1*] \n $\cdots$ [*Input L*] [*Label L*] [*Query Input*]*"*. We term this setting ICL(L) as it corresponds to standard in-context learning. Here we empirically study the effect of label flipping on unlearning success. A comparison of the standard ICL approach (Figure 6, left), where the label of the point we aim to remove is kept unchanged, with our proposed ICULmethod (Figure 6, right) illustrates that label flipping is a crucial factor that pushes the ICUL curve closer to the random guessing benchmark. This finding highlights the essential role of label flipping in successful unlearning and complements recent studies that explore its significance in standard ICL (Min et al., 2022; Wei et al., 2023). These studies suggest that only large-scale language models' test accuracy is affected by fully randomized label flipping. Complementing these findings, our results suggest that smaller LLMs can adjust their predictions to mimic an output distribution that has never encountered the points aimed for removal before.

Dependence on forget point. The last key aspect to consider is whether ICUL requires dependence on the points to be forgotten. To analyze this aspect, the unlearning point from step 1 is substituted with a randomly selected training point paired with a different label, resulting in the subsequent prompt: *"*[*Random Train Input*] [*Different Label*] \n [*Input 1*] [*Label 1*] \n $\cdots$ [*Input L*] [*Label L*] [*Query Input*]*"*. We call this setting Random ICUL(L). Therefore, we examine whether the point in-

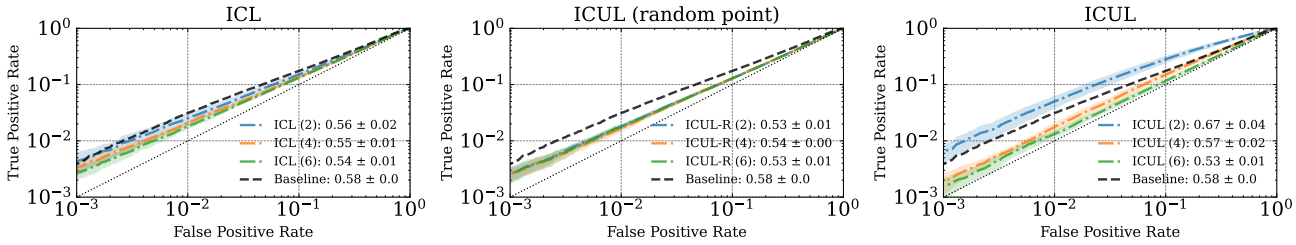


Figure 6. **Towards understanding ICUL.** We plot LiRA−Forget performances using log-scaled AUC curves for a Bloom 1.1B LLM finedtuned on the SST-2 dataset. Here we consider 1 deletion request. The different experiments are described in more detail in Section 5.5. **Left**: Standard in-context learning with correct label. **Center**: Here we follow the ICUL construction, but where the forget point is exchanged for a random point from the data distribution. **Right**: Our suggested ICUL as described in Section 4. Closer proximity to the dotted diagonal, symbolizing the Benchmark, indicates superior performance.

tended for deletion needs to be part of the prompt. Evidence supporting this requirement is displayed by comparing the middle and right plots in Figure 6. This comparison highlights that in the low FPR regime at or below 10^{-2}, our proposed ICUL method substantially surpasses the ICUL that uses a random point.

Taken together, these results show that it is not merely providing examples in-context that results in the measured unlearning, it is the fact that we specifically change the label of the point(s) in question, and then pad the context with 2 to 6 examples with the correct label.

6. Conclusion

In this work, we presented a novel class of unlearning algorithms for LLMs that unlearn even without access to the model parameters. Our method effectively creates a model output distribution that mimics the scenario where a particular point was never part of the model's training dataset. Our algorithm for ICUL creates prompts comprising data points targeted for removal, their changed labels, as well as other accurately labeled instances, which are then provided as inputs to the LLM during inference. In order to evaluate our unlearning algorithm, we extend prior work on membership inference and measuring forgetting to empirically measure unlearning using a likelihood-ratio based test we call LiRA−Forget.

Our empirical results suggest that ICUL reliably removes the influence of training points on the model since LiRA−Forget cannot reliably distinguish between held out points and training points that were subsequently unlearned from the model. Because of its practical appeal and the novelty of our approach, this work establishes a novel perspective on the field of machine unlearning.

Finally, our work offers several questions for exploration:

- **Generalizing ICUL to more complex tasks:** The effective use and evaluation of ICUL for tasks such as open ended data generation remains unclear and requires further investigation.
- **Enabling larger deletion requests:** The current context design makes handling larger deletion requests infeasible, presenting an opportunity for future work.
- **Reducing test time runtime:** As deletion requests increase in size, our ICUL prompts become longer, which consequently increases the test time runtime. To address this, more sophisticated prompt strategies are needed to maintain unlearning efficacy without being dependent on prompt length.
- **Improving prompt designs:** Future research could explore savvier prompts to mitigate large accuracy drops. In Section 5.4, we have demonstrated that ICUL can be adapted to question-answering tasks, but for larger deletion requests like 10, we observed that the model accuracy post unlearning dropped by more than 15%.
- **Prompt inversion attacks against ICUL:** LLMs are vulnerable to various types of attacks, including extracting output layers from the logits, and reconstructing some of the prompts from the output logits, as demonstrated in recent research (Morris et al., 2023; Nasr et al., 2023). In particular, Morris et al. (2023) suggest that in-context learning methods create new privacy risks; for example, it is possible to "steal" the prompt (and thus in-context examples) and leak personal information. An important question to consider is whether ICUL is susceptible to prompt stealing attacks, and if so, how ICUL can be modified to mitigate this risk.
- **Developing more practical unlearning tests:** In Section 3.2, we introduced LiRA−Forget, a new likelihood ratio-based test for assessing unlearning success. This test requires training multiple shadow models, which can be computationally infeasible if the models are large and require significant GPU resources. Future research should focus on developing more efficient unlearning tests that can be more easily applied to (finetuned) large language models with billions of parameters.

Acknowledgements

This work is supported in part by the NSF awards IIS-2008461, IIS-2040989, IIS-2238714, the AI2050 program at Schmidt Sciences, and faculty research awards from Google, JPMorgan, Harvard Data Science Initiative, and the Digital, Data, and Design (Dˆ3) Institute at Harvard. The views expressed here are those of the authors and do not reflect the official policy or position of the funding agencies

Impact Statement

Our proposed In-Context Unlearning method presents a valuable tool for decision makers seeking to make a fine-tuned model behave as if specific data points had been removed from an LLM. Our method has applications across various domains relying on algorithmic decision-making using classification algorithms, including healthcare, education, insurance, credit scoring, recruitment, and criminal justice.

Recognizing the potential of `ICUL` comes with a responsibility to understand its nuances. While offering unparalleled capabilities, like any unlearning algorithm, `ICUL` is not without limitations. Failures in unlearning may manifest under specific hyperparameter configurations, such as setting the number of correctly labeled examples (L) too low. To maximize `ICUL`'s efficacy, decision-makers must grasp both its strengths and weaknesses. Therefore, we advocate for the complementary use of our proposed `LiRA-Forget` test, providing a robust mechanism for validating unlearning outcomes.

In navigating the complex terrain of algorithmic decision-making, our work not only presents a breakthrough method but also equips practitioners with the insights and tools necessary for responsible and informed unlearning of specific information in LLM based classification and question answering tasks.

References

Belrose, N., Schneider-Joseph, D., Ravfogel, S., Cotterell, R., Raff, E., and Biderman, S. Leace: Perfect linear concept erasure in closed form. *arXiv preprint arXiv:2306.03819*, 2023.

Biega, A. J., Potash, P., Daumé, H., Diaz, F., and Finck, M. Operationalizing the legal principle of data minimization for personalization. In *ACM(43) SIGIR '20*, pp. 399–408, 2020.

Borisov, V., Seßler, K., Leemann, T., Pawelczyk, M., and Kasneci, G. Language models are realistic tabular data generators. In *International Conference on Learning Representations (ICLR)*, 2023.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv:2303.12712*, 2023.

Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramer, F. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897–1914. IEEE, 2022.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. Quantifying memorization across neural language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Cook, R. D. and Weisberg, S. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 22(4):495–508, 1980.

Ding, J., Ma, S., Dong, L., Zhang, X., Huang, S., Wang, W., Zheng, N., and Wei, F. Longnet: Scaling transformers to 1,000,000,000 tokens. *arXiv preprint arXiv:2307.02486*, 2023.

Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., and Sui, Z. A survey for in-context learning. *arXiv:2301.00234*, 2023.

Foster, J., Schoepf, S., and Brintrup, A. Fast machine unlearning without retraining through selective synaptic dampening, 2023.

Garg, S., Tsipras, D., Liang, P. S., and Valiant, G. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Ginart, A., Guan, M. Y., Valiant, G., and Zou, J. Making ai forget you: Data deletion in machine learning. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.

Goel, S., Prabhu, A., Sanyal, A., Lim, S.-N., Torr, P., and Kumaraguru, P. Towards adversarial evaluations for inexact machine unlearning. *arXiv preprint arXiv:2201.06640*, 2022.

Golatkar, A., Achille, A., and Soatto, S. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020a.

Golatkar, A., Achille, A., and Soatto, S. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. *arXiv:2003.02960*, 2020b.

Golatkar, A., Achille, A., Ravichandran, A., Polito, M., and Soatto, S. Mixed-privacy forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 792–801, 2021.

Goldsteen, A., Ezov, G., Shmelkin, R., Moffie, M., and Farkash, A. Data minimization for gdpr compliance in machine learning models. *AI and Ethics*, pp. 1–15, 2021.

Graves, L., Nagisetty, V., and Ganesh, V. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Guo, C., Goldstein, T., Hannun, A., and Van Der Maaten, L. Certified data removal from machine learning models. In *International Conference on Machine Learing (ICML)*, 2019.

Hayes, J., Shumailov, I., Triantafillou, E., Khalifa, A., and Papernot, N. Inexact unlearning needs more careful evaluations to avoid a false sense of privacy, 2024.

Henderson, P., Li, X., Jurafsky, D., Hashimoto, T., Lemley, M. A., and Liang, P. Foundation models and fair use. *arXiv:2303.15715*, 2023.

Huang, Y. and Canonne, C. L. Tight bounds for machine unlearning via differential privacy. *arXiv:2309.00886*, 2023.

Izzo, Z., Anne Smart, M., Chaudhuri, K., and Zou, J. Approximate data deletion from machine learning models. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

Jagielski, M., Thakkar, O., Tramer, F., Ippolito, D., Lee, K., Carlini, N., Wallace, E., Song, S., Thakurta, A., Papernot, N., et al. Measuring forgetting of memorized training examples. In *International Conference on Learning Representations (ICLR)*, 2023.

Jang, J., Yoon, D., Yang, S., Cha, S., Lee, M., Logeswaran, L., and Seo, M. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

Jia, J., Liu, J., Ram, P., Yao, Y., Liu, G., Liu, Y., Sharma, P., and Liu, S. Model sparsity can simplify machine unlearning, 2024.

Kurmanji, M., Triantafillou, P., and Triantafillou, E. Towards unbounded machine unlearning. *arXiv preprint arXiv:2302.09880*, 2023.

Leemann, T., Pawelczyk, M., and Kasneci, G. Gaussian membership inference privacy. In *Advances in neural information processing systems (NeurIPS)*, 2023.

Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., and Chen, W. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 2022.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55, 2023.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, 2022.

Morris, J. X., Zhao, W., Chiu, J. T., Shmatikov, V., and Rush, A. M. Language model inversion. *arXiv preprint arXiv:2311.13647*, 2023.

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Wallace, E., Tramèr, F., and Lee, K. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.

Neel, S., Roth, A., and Sharifi-Malvajerdi, S. Descent-to-delete: Gradient-based methods for machine unlearning. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory (ALT)*, 2021.

Neyman, J. and Pearson, E. S. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

OAG, C. Ccpa regulations: Final regulation text. *Office of the Attorney General, California Department of Justice*, 2021.

Pang, B. and Lee, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*, 2005.

Pawelczyk, M., Leemann, T., Biega, A., and Kasneci, G. On the trade-off between actionable explanations and the right to be forgotten. In *International Conference on Learning Representations (ICLR)*, 2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. SQuAD: 100,000+ questions for machine comprehension of text. In Su, J., Duh, K., and Carreras, X. (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Ravfogel, S., Twiton, M., Goldberg, Y., and Cotterell, R. D. Linear adversarial concept erasure. In *International Conference on Machine Learning*, pp. 18400–18421. PMLR, 2022a.

Ravfogel, S., Vargas, F., Goldberg, Y., and Cotterell, R. Adversarial concept erasure in kernel space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6034–6055, 2022b.

Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., Tow, J., Rush, A. M., Biderman, S., Webson, A., Ammanamanchi, P. S., Wang, T., Sagot, B., Muennighoff, N., del Moral, A. V., Ruwase, O., Bawden, R., Bekman, S., McMillan-Major, A., Beltagy, I., Nguyen, H., Saulnier, L., Tan, S., Suarez, P. O., Sanh, V., Laurençon, H., Jernite, Y., Launay, J., Mitchell, M., Raffel, C., Gokaslan, A., Simhi, A., Soroa, A., Aji, A. F., Alfassy, A., Rogers, A., Nitzav, A. K., Xu, C., Mou, C., Emezue, C., Klamm, C., Leong, C., van Strien, D., Adelani, D. I., Radev, D., Ponferrada, E. G., Levkovizh, E., Kim, E., Natan, E. B., De Toni, F., Dupont, G., Kruszewski, G., Pistilli, G., Elsahar, H., Benyamina, H., Tran, H., Yu, I., Abdulmumin, I., Johnson, I., Gonzalez-Dios, I., de la Rosa, J., Chim, J., Dodge, J., Zhu, J., Chang, J., Frohberg, J., Tobing, J., Bhattacharjee, J., Almubarak, K., Chen, K., Lo, K., Von Werra, L., Weber, L., Phan, L., allal, L. B., Tanguy, L., Dey, M., Muñoz, M. R., Masoud, M., Grandury, M., Šaško, M., Huang, M., Coavoux, M., Singh, M., Jiang, M. T.-J., Vu, M. C., Jauhar, M. A., Ghaleb, M., Subramani, N., Kassner, N., Khamis, N., Nguyen, O., Espejel, O., de Gibert, O., Villegas, P., Henderson, P., Colombo, P., Amuok, P., Lhoest, Q., Harliman, R., Bommasani, R., López, R. L., Ribeiro, R., Osei, S., Pyysalo, S., Nagel, S., Bose, S., Muhammad, S. H., Sharma, S., Longpre, S., Nikpoor, S., Silberberg, S., Pai, S., Zink, S., Torrent, T. T., Schick, T., Thrush, T., Danchev, V., Nikoulina, V., Laippala, V., Lepercq, V., Prabhu, V., Alyafeai, Z., Talat, Z., Raja, A., Heinzerling, B., Si, C., Salesky, E., Mielke, S. J., Lee, W. Y., Sharma, A., Santilli, A., Chaffin, A., Stiegler, A., Datta, D., Szczechla, E., Chhablani, G., Wang, H., Pandey, H., Strobelt, H., Fries, J. A., Rozen, J., Gao, L., Sutawika, L., Bari, M. S., Al-shaibani, M. S., Manica, M., Nayak, N., Teehan, R., Albanie, S., Shen, S., Ben-David, S., Bach, S. H., Kim, T., Bers, T., Fevry, T., Neeraj, T., Thakker, U., Raunak, V., Tang, X., Yong, Z.-X., Sun, Z., Brody, S., Uri, Y., Tojarieh, H., Roberts, A., Chung, H. W., Tae, J., Phang, J., Press, O., Li, C., Narayanan, D., Bourfoune, H., Casper, J., Rasley, J., Ryabinin, M., Mishra, M., Zhang, M., Shoeybi, M., Peyrounette, M., Patry, N., Tazi, N., Sanseviero, O., von Platen, P., Cornette, P., Lavallée, P. F., Lacroix, R., Rajbhandari, S., Gandhi, S., Smith, S., Requena, S., Patil, S., Dettmers, T., Baruwa, A., Singh, A., Cheveleva, A., Ligozat, A.-L., Subramonian, A., Névéol, A., Lovering, C., Garrette, D., Tunuguntla, D., Reiter, E., Taktasheva, E., Voloshina, E., Bogdanov, E., Winata, G. I., Schoelkopf, H., Kalo, J.-C., Novikova, J., Forde, J. Z., Clive, J., Kasai, J., Kawamura, K., Hazan, L., Carpuat, M., Clinciu, M., Kim, N., Cheng, N., Serikov, O., Antverg, O., van der Wal, O., Zhang, R., Zhang, R., Gehrmann, S., Pais, S., Shavrina, T., Scialom, T., Yun, T., Limisiewicz, T., Rieser, V., Protasov, V., Mikhailov, V., Pruksachatkun, Y., Belinkov, Y., Bamberger, Z., Kasner, Z., Rueda, A., Pestana, A., Feizpour, A., Khan, A., Faranak, A., Santos, A., Hevia, A., Unldreaj, A., Aghagol, A., Abdollahi, A., Tammour, A., HajiHosseini, A., Behroozi, B., Ajibade, B., Saxena, B., Ferrandis, C. M., Contractor, D., Lansky, D., David, D., Kiela, D., Nguyen, D. A., Tan, E., Baylor, E., Ozoani, E., Mirza, F., Ononiwu, F., Rezanejad, H., Jones, H., Bhattacharya, I., Solaiman, I., Sedenko, I., Nejadgholi, I., Passmore, J., Seltzer, J., Sanz, J. B., Fort, K., Dutra, L., Samagaio, M., Elbadri, M., Mieskes, M., Gerchick, M., Akinlolu, M., McKenna, M., Qiu, M., Ghauri, M., Burynok, M., Abrar, N., Rajani, N., Elkott, N., Fahmy, N., Samuel, O., An, R., Kromann, R., Hao, R., Alizadeh, S., Shubber, S., Wang, S., Roy, S., Viguier, S., Le, T., Oyebade, T., Le, T., Yang, Y., Nguyen, Z., Kashyap, A. R., Palasciano, A., Callahan, A., Shukla, A., Miranda-Escalada, A., Singh, A., Beilharz, B., Wang, B., Brito, C., Zhou, C., Jain, C., Xu, C., Fourrier, C., Periñán, D. L., Molano, D., Yu, D., Manjavacas, E., Barth, F., Fuhrimann, F., Altay, G., Bayrak, G., Burns, G., Vrabec, H. U., Bello, I., Dash, I., Kang, J., Giorgi, J., Golde, J., Posada, J. D., Sivaraman, K. R., Bulchandani, L., Liu, L., Shinzato, L., de Bykhovetz, M. H., Takeuchi, M., Pàmies, M., Castillo, M. A., Nezhurina, M., Sänger, M., Samwald, M., Cullan, M., Weinberg, M., De Wolf, M., Mihaljcic, M., Liu, M., Freidank, M., Kang, M., Seelam, N., Dahlberg, N., Broad, N. M., Muellner, N., Fung, P., Haller, P., Chandrasekhar, R., Eisenberg, R., Martin, R., Canalli, R., Su, R., Su, R., Cahyawijaya, S., Garda, S., Deshmukh, S. S., Mishra, S., Kiblawi, S., Ott, S., Sang-aroonsiri, S., Kumar, S., Schweter, S., Bharati, S., Laud, T., Gigant, T., Kainuma, T., Kusa, W., Labrak, Y., Bajaj, Y. S., Venkatraman, Y., Xu, Y., Xu, Y., Xu, Y., Tan, Z., Xie, Z., Ye, Z., Bras, M., Belkada, Y., and Wolf, T. Bloom: A 176b-parameter open-access multilingual language model. *arXiv:2211.05100*, 2022.

Sekhari, A., Acharya, J., Kamath, G., and Suresh, A. T. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems*, 2021.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.

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

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tsybakov, A. B. *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 1st edition, 2008. ISBN 0387790519.

Union, E. Regulation (eu) 2016/679 of the european parliament and of the council. *Official Journal of the European Union*, 2016.

Voigt, P. and Von dem Bussche, A. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10:3152676, 2017.

Wang, L., Chen, T., Yuan, W., Zeng, X., Wong, K.-F., and Yin, H. Kga: A general machine unlearning framework based on knowledge gap alignment. *arXiv preprint arXiv:2305.06535*, 2023.

Wei, J., Wei, J., Tay, Y., Tran, D., Webson, A., Lu, Y., Chen, X., Liu, H., Huang, D., Zhou, D., et al. Larger language models do in-context learning differently. *arXiv:2303.03846*, 2023.

Wu, Y., Dobriban, E., and Davidson, S. Deltagrad: Rapid retraining of machine learning models. In *International Conference on Machine Learning (ICML)*, 2020.

Zhang, R. and Zhang, S. Rethinking influence functions of neural networks in the over-parameterized regime. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In *Advances in neural information processing systems (NeurIPS)*, volume 28, 2015.

A. Reproducibility Statement

Overall compute requirements. To run all experiments[1], we need to save the model weights of the shadow models required for `LiRA-Forget`. To reproduce all our experiments, set 1500 GB of storage aside. All experiments that use Bloom models are run using Nvidia Tesla V100 GPUs (32 GB RAM). For model finetuning on the Llama2 7B LLM, we use one A100 GPU (80 GB RAM); note that during finetuning we update *all* 7B model parameters. For unlearning via `ICUL`, we use Tesla V100 GPUs (32 GB RAM).

Experimentwise compute requirements. Next, we provide an overview on the number of GPU hours to reproduce our experiments.

1. **Evaluating unlearning** (Figure 5): For every dataset, we finetuned 10 LLMs for one epoch for every forget set size (1, 5, 10, 20). This is required to run `LiRA-Forget`. Finetuning one model roughly takes 1 GPU hour; finetuning all models roughly takes 40 GPU hours per dataset. Including hyperparameter search, we ran the unlearning procedures on *all* 25000 points. First, for `ICUL` we ran inference across 3 context length configurations across 40 models and each run took 2 hours on average. Across all three data sets, this amount to a total of 600 GPU hours. For `GA`, the situation was similar. We ran the unlearning procedure for 4 learning rate configurations across 40 models and each run took roughly 2 hours. This amount to a total of 600 GPU hours. In total, to reproduce this experiment requires 1800 GPU hours of compute, or roughly 75 GPU days.

2. **Evaluating `ICUL` sensitivity across model sizes** (Figure 3): For the SST-2 dataset and the forget set size of 10, we finetuned 10 LLMs per model size. This took roughly 5 hours for the smaller 560M models, 10 hours for the 1.1B model and 14 hours for the 3B model. For each of these model, we ran inference 10 times. For the smaller 560M models, inference took roughly 2 hours per model, while inference took approx. 3 hours for the 1.1B model and roughly 5 hours for the 3B model. In total, this experiment will roughly take 130 GPU hours to reproduce.

3. **`ICUL` demonstrates effectiveness on state-of-the-art LLMs** (Figure 4): For the SST-2 dataset and the forget set size of 1, 10 and 20 we finetuned 10 LLMs per forget set size. Finetuning Llama-2 (7B) took roughly 48 hours per forget set size on one A100 GPU (80GB). Performing 10 inference runs on the three models required another 8 hours per run on average on the V100 GPU (32 GB). In total, this experiment roughly required 385 GPU hours.

B. Details on the Machine Unlearning Evaluation

Operationalizing the Likelihood-ratio Audit. Operationalizing the likelihood ratio test from (2) requires access to the distribution of losses under the null and alternative hypotheses. While analytical solutions are usually not available, we can readily get large samples from these two distributions. In an ideal scenario, this entails that we would need to fit as many re-train models and unlearned models as possible for every forget set of interest. Since this approach becomes computationally too burdensome, we use the following approximation:

Approximating the distributions under H_0 and H_1 from equation (1). Here we adapt the sample splitting procedure first introduced by Carlini et al. (2022) to forget sets with sizes $J = \{1, 5, 10, 20\}$. We train K shadow models on random samples from the data distribution $\mathcal{D}$ so that a fraction p of these models are trained on the forget set $S_f = \{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1}^{J}$, and a fraction $(1-p)$ are not. In particular, we train shadow models on $K = 10$ subsets of $\mathcal{D}$ so that each forget set $S_f \in D$ appears in $K \cdot p$ subsets. This approach has the advantage that the same K shadow models can be used to estimate the likelihood-ratio test for all the forget sets. Finally, we fit the parameters of two Gaussian distributions to the confidence scores of the retain models and the unlearned models on S_f. Across all experiments, we use $p = 0.5$.

Model losses. Instead of using the actual losses, we follow Carlini et al. (2022) and compute model confidences as $\phi(f(\mathbf{x}), \mathbf{y}) = \log(f(\mathbf{x})_y) - \log(\sum_{y'} f(\mathbf{x})_{y'})$ which the authors show yields the strongest empirical attack performance. This score compares the confidence the model assigns to the true class (e.g., 'positive') with the confidences the model assigns to all other classes (i.e., all other words from the approximately 250680 dimensional vocabulary). The higher the score is the more confident the model is in the correct prediction.

[1]We release our code at: https://github.com/MartinPawel/In-Context-Unlearning.

# Deletions	GA	ICUL
1	64.80	29.90
5	64.79	30.57
10	68.91	31.92
20	68.90	34.16

(a) Maximum GPU RAM utilization measured in GB for GA and ICUL across different numbers of deletions.

# Deletions	GA	ICUL
1	1.44	0.00
5	3.41	0.00
10	7.60	0.00
20	15.94	0.00

(b) Model update wall clock time measured in seconds for GA and ICUL across different numbers of deletions.

Table 2. Computational resources required to update Llama2 (7B) on the SST-2 dataset across unlearning methods.

# Inference Runs	GA	ICUL (1 Deletion)	ICUL (5 Deletions)	ICUL (10 Deletions)	ICUL (20 Deletions)
1	0.40	0.42	0.57	0.99	1.64
5	2.01	2.10	2.85	5.00	8.20
10	4.03	4.20	5.70	10.00	16.40
20	8.00	8.40	11.40	20.00	32.80

Table 3. Inference run times measured in seconds for GA and ICUL across different numbers of deletions.

C. Additional Results

C.1. Additional Empirical Comparisons: Compute Times and Memory Requirements

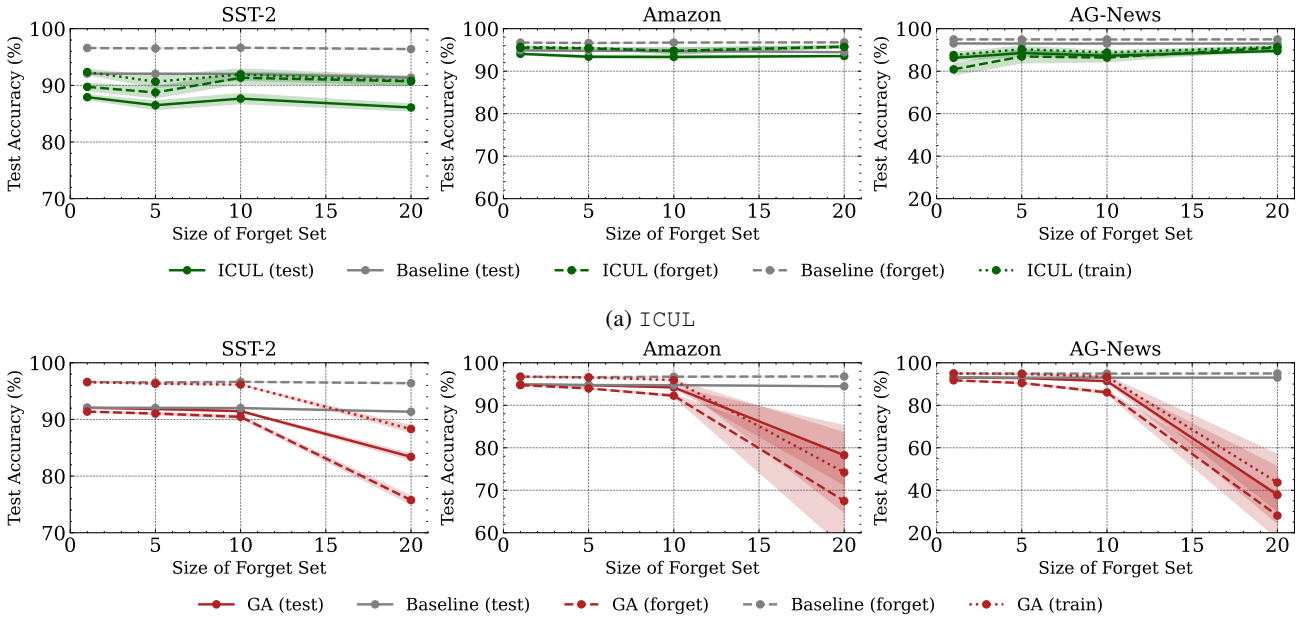
To illustrate an extensive computational cost comparison between GA and ICUL, we present evaluate memory cost and storage requirements for a Llama2 (7B) LLM on the SST-2 dataset. It is important to note that ICUL is specifically designed for GPU RAM-constrained compute environments. While increasing GPU RAM to fine-tune may pose challenges for many users, they might be more willing to accept additional computation time to obtain desired results. To provide a more nuanced perspective, we identify three distinct dimensions of computational costs associated with any unlearning method:

- The memory requirement for executing the model update (see Table 2a);
- The computational time required to update the model (see Table 2b);
- The computational time needed to perform inference runs using the updated model (see Table 3).

Looking ahead, further advancements in transformer architectures such as (Ding et al., 2023) are significantly reducing inference times by improving computational efficiency for larger context lengths. In particular, the work by Ding et al. (2023) boasts linear computation complexity in the sequence length without sacrificing performance. This development will likely further enhance the utility of ICUL as compute times for larger contexts decrease.

C.2. Additional Empirical Comparisons: Test Accuracy and Unlearning Efficacy

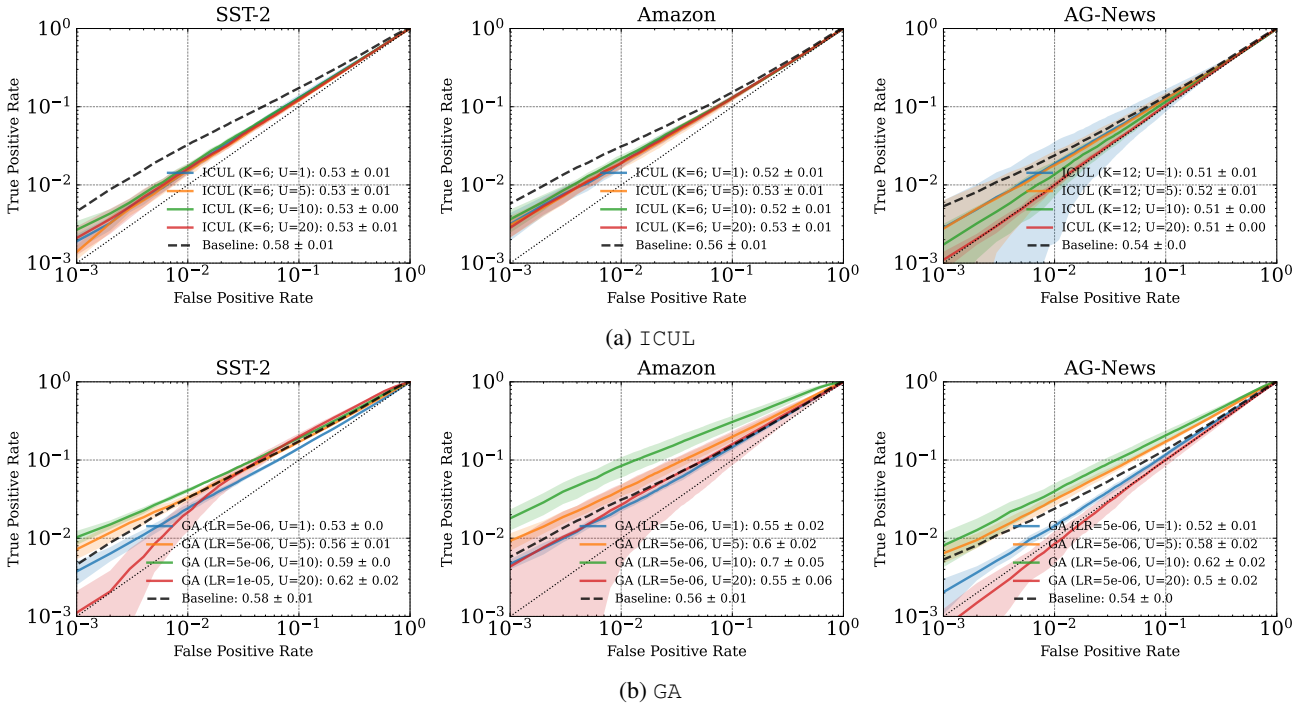
Here we additionally experiment with the Yelp polarity data set that was was originally introduced by Zhang et al. (2015).



(a) ICUL

(b) GA

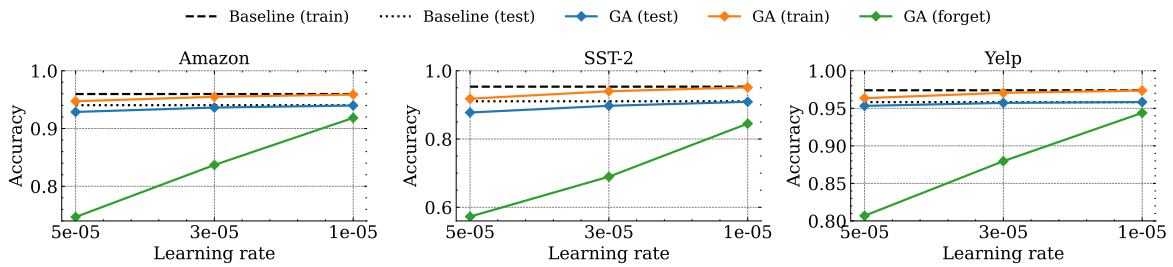
Figure 7. **Classification performance as we vary the size of the forget set**. We report classification accuracy on train, forget and test points across all data sets for the 1.1B Bloom LLM.



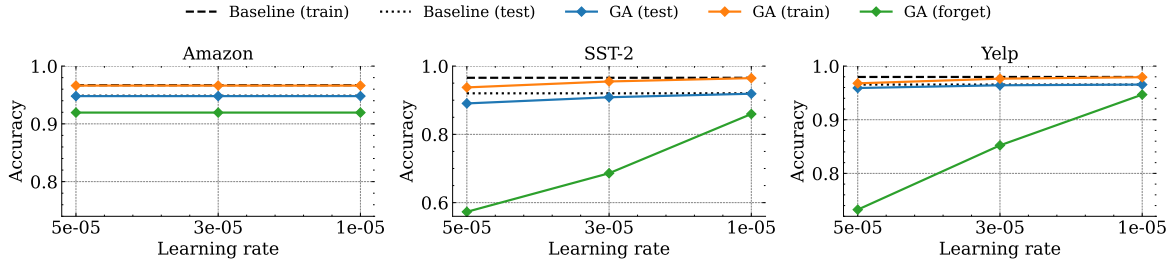
(a) ICUL

(b) GA

Figure 8. **Log scaled AUC curves**. Here we show the complete AUC curves for the most competitive hyperparameters on all data sets for the Bloom 1.1B model for both GA and ICUL.

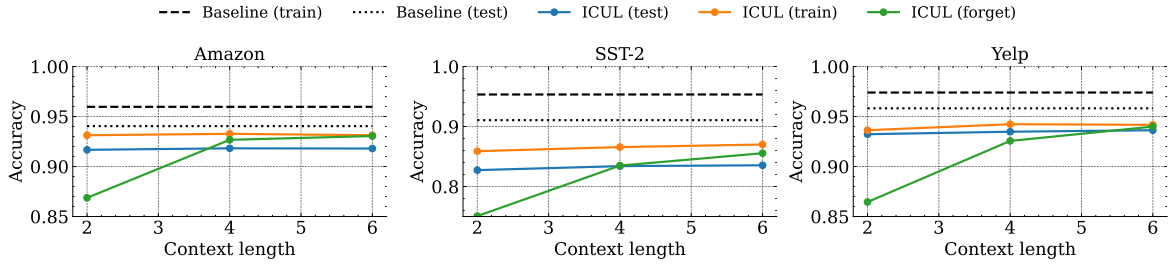


(a) Bloom (560M parameters)

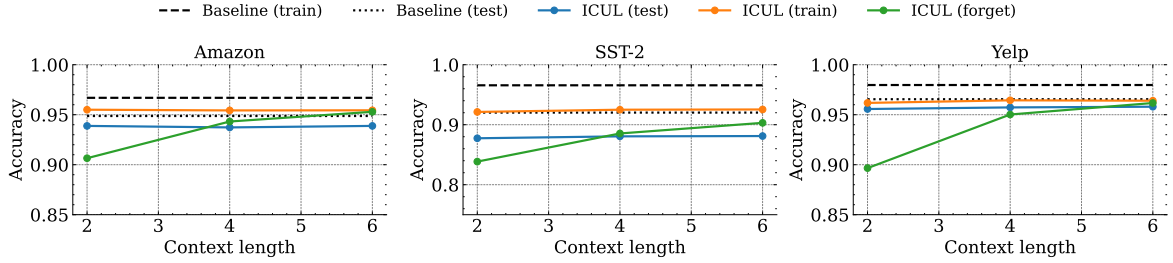


(b) Bloom (1.1B parameters)

Figure 9. **Classification performance as we vary the learning rate for `GA`.** We report classification accuracy on train, forget and test points across all data sets and model sizes. For better readability, ± 1 standard deviation was excluded from the figure.



(a) Bloom (560M parameters)



(b) Bloom (1.1B parameters)

Figure 10. **Classification performance as we vary context length for `ICUL`.** We report classification accuracy on train, forget and test points across all data sets and model sizes. For better readability, ± 1 standard deviation was excluded.