# Learn while Unlearn: An Iterative Unlearning Framework for Generative Language Models

**Haoyu Tang***, **Ye Liu***, **Xukai Liu, Kai Zhang, Yanghai Zhang, Qi Liu, Enhong Chen**
State Key Laboratory of Cognitive Intelligence
University of Science and Technology of China
{haoyu_t,liuyer,chthollylxk,apocalypseh}@mail.ustc.edu.cn;
{kkzhang08,qiliuql,cheneh}@ustc.edu.cn

## Abstract

Recent advancements in machine learning, especially in Natural Language Processing (NLP), have led to the development of sophisticated models trained on vast datasets, but this progress has raised concerns about potential sensitive information leakage. In response, regulatory measures like the EU General Data Protection Regulation (GDPR) have driven the exploration of Machine Unlearning techniques, which aim to enable models to selectively forget certain data entries. While early approaches focused on pre-processing methods, recent research has shifted towards training-based machine unlearning methods. However, many existing methods require access to original training data, posing challenges in scenarios where such data is unavailable. Besides, directly facilitating unlearning may undermine the language model's general expressive ability. To this end, in this paper, we introduce the **I**terative **C**ontrastive **U**nlearning (**ICU**) framework, which addresses these challenges by incorporating three key components. We propose a Knowledge Unlearning Induction module for unlearning specific target sequences and a Contrastive Learning Enhancement module to prevent degrading in generation capacity. Additionally, an Iterative Unlearning Refinement module is integrated to make the process more adaptive to each target sample respectively. Experimental results demonstrate the efficacy of ICU in maintaining performance while efficiently unlearning sensitive information, offering a promising avenue for privacy-conscious machine learning applications.

## 1 Introduction

With the continuous evolution of machine learning, we have witnessed an explosion in the size of models and the breadth of data used for their training. Particularly, the field of NLP has seen remarkable progress, with the advent of advanced Generative Language Models (GLM) such as GPT-4 [1], Claude 3 [4], and Google Gemini [48]. Nonetheless, concurrent studies have illuminated a concerning aspect of these models: the potential for the leakage of sensitive information [12], including but not limited to, phone numbers, email addresses, and other personal data embedded within the training datasets [8, 13]. In response to privacy concerns, the GDPR [53] has codified the "Right To Be Forgotten" (RTBF) [35, 52], leading to the exploration of Machine Unlearning techniques. These techniques aim to make models effectively "forget" certain data entries, treating them as if they were never part of the training dataset.

In the early stage, researchers attempted various pre-processing methods to achieve the goal for forgetting certain data. For instance, Kandpal et al. [29] find that the rate at which language models regenerate training sequence is related to the sequence's count in the training set and that after

*Equal contribution

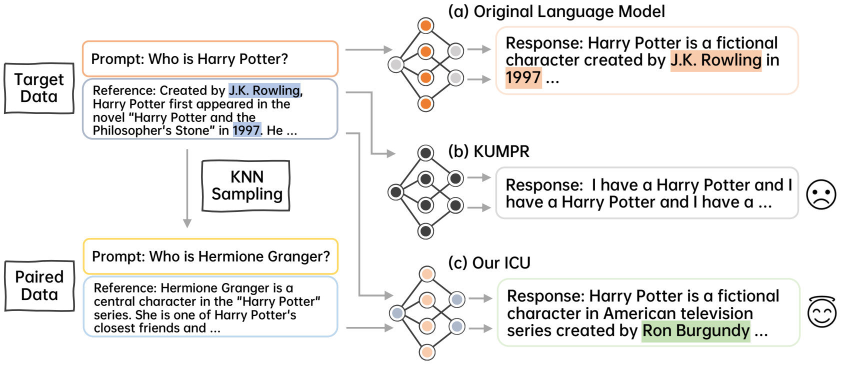
Target Data
Prompt: Who is Harry Potter?
Reference: Created by J.K. Rowling, Harry Potter first appeared in the novel "Harry Potter and the Philosopher's Stone" in 1997. He ...
KNN Sampling
Paired Data
Prompt: Who is Hermione Granger?
Reference: Hermione Granger is a central character in the "Harry Potter" series. She is one of Harry Potter's closest friends and ...
(a) Original Language Model
Response: Harry Potter is a fictional character created by J.K. Rowling in 1997 ...
(b) KUMPR
Response: I have a Harry Potter and I have a Harry Potter and I have a ...
(c) Our ICU
Response: Harry Potter is a fictional character in American television series created by Ron Burgundy ...

Figure 1: Difference among the generated sequences of (a) original model, (b) model unlearned by KUMPR [27] and (c) model unlearned by our method.

applying methods to deduplicate training data, language models are considerably more secure against privacy attacks. However, in practice, such methods are time-consuming and resource-intensive. Besides, they cannot tackle the condition where requests for unlearning come frequently.

In recent years, researchers turns to training-based machine unlearning approaches, which modify the training process itself rather than solely manipulating the data. For example, SISA [7] partitions the original dataset into several non-overlapping shards and then aggregates models trained separately. When handling data deletion, this method only has to retrain the models trained on the affected shards. KGA [54] introduces extra dataset in addition to the original dataset, and finetune the original model based on these datasets. Nonetheless, both these two methods require that training data be available for unlearning. In practice, the large language models may be released without the training data, making such methods infeasible. Knowledge Unlearning for Mitigating Privacy Risks [27] is such a typical method that applies to such scenarios.[2] It reverses the training objective of minimizing the negative log-likelihood for the target tokens and use metrics to determine whether the model has "forgotten" a target sequence.

However, directly reversing the training objective of language models may undermine the overall expressive ability of the language model, not just forgetting the given data. For example, as observed in Figure 1, given a training sample with prefix "*Who is Harry Potter?*" and referenced suffix, we aim to remove the learned knowledge from the language model, such as "*J.K. Rowling*" and "*1997*" in the referenced suffix. For the KUMPR method, although it can successfully forget the key information contained in the reference suffix, the unlearned model loses the expressive ability, thus deriving the repeating sentence "*I have a Harry Potter*". In response to this shortcoming, we propose an Iterative Contrastive Unlearning (ICU) framework, which can preserve the generalization ability while achieving the unlearning objective.

More specifically, our ICU method consists of 3 components: (1) a Knowledge Unlearning Induction (KUI) module, which uses the target to unlearn specific knowledge from the model; (2) a Contrastive Learning Enhancement (CLE) module, which samples paired data from a subset of the training dataset corresponding to the target data and trains the model accordingly; and (3) an Iterative Unlearning Refinement (IUR) module, which filters the target dataset during the unlearning process based on the effectiveness of the extraction attack on each sample. The experimental results and further analysis from various perspectives demonstrate that our proposed method outperforms baselines in terms of performance maintenance and unlearning efficiency.

In brief, our contributions are as follows:

- We make an in-depth study for the unlearning technique of Generative Language Models, specifically for Large Language Models, which has long been ignored.
- We design ICU framework, which comprises Knowledge Unlearning Induction, Contrastive Learning Enhancement and Iterative Unlearning Refinement.

[2] We refer to the method as KUMPR in the following parts, as the authors did not provide a specific name.

- We conduct extensive experiments on three different sized backbones, where the experimental results demonstrate the effectiveness of our proposed method. Our code is available at https://github.com/himalalps/ICU.

## 2 Related Work

### 2.1 Machine Unlearning

Introduced by Cao and Yang [11], machine unlearning, aims at protecting machine learning models from extraction attacks, involves the process of removing specific data from a model in a manner that ensures the data appears as though it were never part of the training set. Conventional approaches [37] of excluding specific data from training datasets and subsequently retraining the model are highly time-consuming and resource-intensive, making it impractical for contemporary deep neural networks. Similar methods [7, 34] that involve retraining also fail to adequately address the issue, particularly when dealing with a large number of deletion requests or limited access to comprehensive datasets. Researchers have investigated approximate unlearning techniques to address the limitations of exact machine unlearning [21, 24, 36]. One such technique is data pre-processing, which efficiently identifies and removes sensitive information before it is incorporated into the model. Kandpal et al. [29] utilize this method on structured private data, including phone numbers and medical records, and found it to be highly effective. However, challenges emerge with less structured data, as pre-processing techniques may inadequately remove all sensitive information [10] and pre-processing alone cannot comprehensively address deletion demands [8].

Recent researches [27, 30, 42, 57] have concentrated on fine-tuning language models to address challenges in machine unlearning. Jang et al. [27] introduce an innovative method by reversing the conventional training objective, aiming to maximize, rather than minimize, the negative log-likelihood of tokens designated for forgetting. Other recent methodologies [14, 19, 23, 30, 54] in this field use diverse techniques such as knowledge gap alignment and reinforcement learning. Despite offering promising solutions to machine unlearning challenges, these methods often exhibit high complexity and computational cost, rendering their practical implementation both time-consuming and intricate. For instance, Gu et al. [23] leverage second-order information (Hessian) to provide stronger guarantees for data removal and maintain model utility, but extensive computational resources are required for Hessian approximation.

### 2.2 Language Models

A language model is a computer program or algorithm that is designed to understand, generate, and predict human language [61]. The concept of machine unlearning for language models has gained increased attention in recent years [27, 30, 42, 57].

Although traditional language models such as rule-based models [55] or statistical models [9] can generate outputs resembling human language, these outputs do not perfectly align with the training dataset. Typical neural networks utilized in NLP, such as Recurrent Neural Network (RNN) [56], and its derivatives like Long Short-Term Memory (LSTM) [25], encounter limitations due to their sequential processing architecture. This sequential structure leads to significant computational resources and time required for training, which hinders the model's scalability and efficiency. As a result, they often struggle to display a high level of proficiency in retaining the information from the training dataset [41].

The introduction of the transformer architecture [51] has revolutionized the realm of NLP. Built upon the self-attention mechanism, transformers facilitate the effective capture of contextual relationships within textual data. Subsequent advancements have diversified transformer-based approaches into three primary categories: encoder-only models, exemplified by BERT [16]; decoder-only models, typified by GPT [43]; and encoder-decoder models such as T5 [44]. Decoder-only models like GPT-4 [1], Claude 3 [4], and others [3, 48, 50] have demonstrated exceptional performance across various NLP tasks. However, this success raises privacy concerns, as these models have the potential to leak sensitive information from their training data. Moreover, alongside the enhancement in model capacity, there has been a corresponding increase in the demand for training data and computational resources [1, 4, 26, 38]. This escalation presents challenges for researchers working on machine unlearning in these advanced models, as it requires substantial effort and resources.

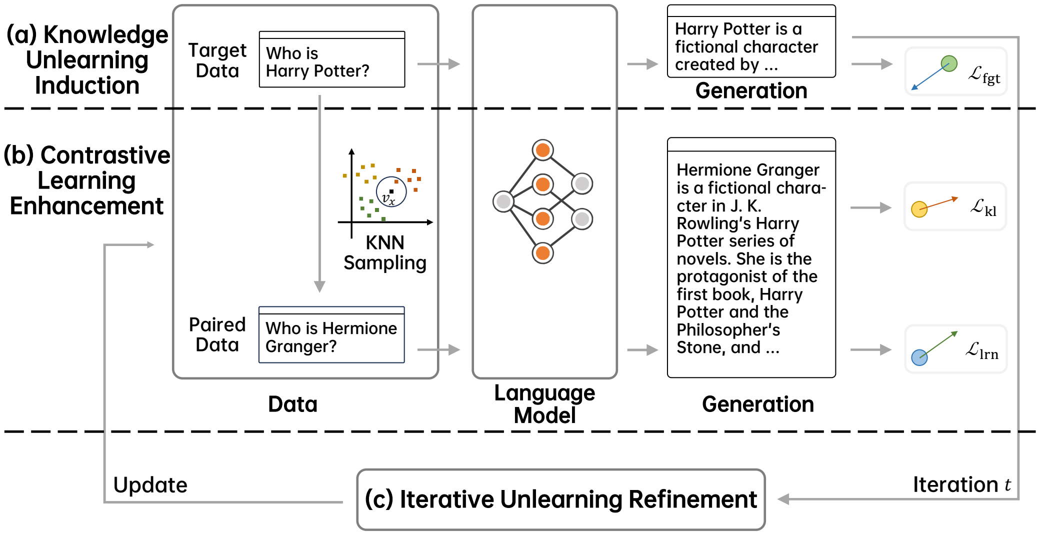
(a) Knowledge Unlearning Induction
Target Data
Who is Harry Potter?
Harry Potter is a fictional character created by ...
Generation
$\mathcal{L}_{\text{fgt}}$
(b) Contrastive Learning Enhancement
$v_x$
KNN Sampling
Hermione Granger is a fictional character in J. K. Rowling's Harry Potter series of novels. She is the protagonist of the first book, Harry Potter and the Philosopher's Stone, and ...
$\mathcal{L}_{\text{kl}}$
$\mathcal{L}_{\text{lrn}}$
Paired Data
Who is Hermione Granger?
Data
Language Model
Generation
Update
(c) Iterative Unlearning Refinement
Iteration $t$

Figure 2: The structure of Iterative Contrastive Unlearning framework. It consists of three parts: (a) Knowledge Unlearning Induction (KUI), (b) Contrastive Learning Enhancement (CLE), and (c) Iterative Unlearning Refinement (IUR).

# 3 Method

## 3.1 Problem Statement

Given the data and model $\{D, D_{fgt}, f_\theta\}$, where $D = \{\boldsymbol{x}_i\}_{i=1}^{N}$ is a small pre-training corpora with $N$ pieces of data, $D_{fgt} = \{\boldsymbol{x}_i^{fgt}\}_{i=1}^{M}$ is the collection of $M$ pieces of data in $D$ to be forgotten and $f_\theta$ is the orginal language model with its parameters denoted as $\theta$, machine unlearning is to modify the parameters $\theta$ of the model $f_\theta$ in such a way that it minimizes the retention of previously learned information about $D_{fgt}$ while still maintaining desirable model performance and satisfying the specified constraints.

## 3.2 Model Overview

We propose a novel Iterative Contrastive Unlearning framework as shown in Figure 2. Aiming to delve into the unlearning process specifically tailored for decoder-only models, we seek to address the challenges of mitigating the memorization of sensitive information while preserving the models' language generation capabilities. In addition to Knowledge Unlearning Induction, which trains the model to forget target sequences, we introduce two supplementary modules. The Contrastive Learning Enhancement module uses specially selected data to maintain overall performance during unlearning training. Additionally, the Iterative Unlearning Refinement module updates the data to be forgotten iteratively, preventing over-unlearning and significant performance degradation.

## 3.3 Knowledge Unlearning Induction

For a sample $\boldsymbol{x}^{fgt} \in D_{fgt}$, the sequence of tokens can be denoted as $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$. Following Jang et al. [27], we *negate* the original negative log-likelihood of the target token sequences to induce the model to forget these sequences. The unlearning objective is computed by the following function:

$$\mathcal{L}_{fgt} = \sum_{t=1}^{T} \log P_\theta(x_t^{fgt} | x_{<t}^{fgt}) \tag{1}$$

where $x_{<t} = (x_1, x_2, \ldots, x_{t-1})$ denotes the first $t$ tokens of the sequence, and $P_\theta(x_t | x_{<t})$ represents the conditional probability of predicting the next token $x_t$ given $x_{<t}$, with $\theta$ being the model parameters.

### 3.4 Contrastive Learning Enhancement

To maintain the stable generation capacity of the language model during unlearning, we propose simultaneously training the model on analogous data. This method ensures that the model forgets specific information without substantially reducing its ability to recognize and generate similar data patterns.

**KNN Sampling** First, we compute the sentence embedding $v$ for all samples in $D$ using a pre-trained sentence transformer $f_s$. For each sample $\boldsymbol{x}^{fgt}$ in $D_{fgt}$ with its embedding $v_x^{fgt} = f_s(\boldsymbol{x}^{fgt})$, we employ K-Nearest Neighbors (KNN) [18] to identify the nearest ($K = 1$) embedding $\hat{v_x}$ and the corresponding sample $\hat{\boldsymbol{x}}$. All $\hat{\boldsymbol{x}}$ are then collected to form $D_{lrn}$.

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x} \in D \setminus D_{fgt}}{\operatorname{argmin}} \; dis(v_x, \hat{v_x}) \tag{2}$$

where $dis(\cdot)$ is the cosine similarity function used in KNN search.

**Learning Enhancement** Subsequently, we have $\hat{\boldsymbol{x}} = \boldsymbol{x}^{lrn} \in D_{lrn}$ which is the paired data of $\boldsymbol{x}^{fgt}$. In contrast to the unlearning objective, we force the model to learn data patterns from the paired token sequences using negative log-likelihood, which can be formulated as follows:

$$\mathcal{L}_{lrn} = -\sum_{t=1}^{T} \log P_\theta(x_t^{lrn} | x_{<t}^{lrn}) \tag{3}$$

Additionally, we apply Kullback-Leibler (KL) divergence [33] to guide the model in learning the original model's distribution for data intended to be retained following Yao et al. [57]:

$$\mathcal{L}_{kl} = \sum_{t=1}^{T} KL[P_{\theta_0}(x_t^{lrn} | x_{<t}^{lrn}) || P_\theta(x_t^{lrn} | x_{<t}^{lrn})] \tag{4}$$

where $\theta_0$ denotes the parameters of the original model.

### 3.5 Iterative Unlearning Refinement

In contrast to conventional machine learning techniques, validating the efficacy of unlearning poses challenges in identifying a suitable validation set since $D_{fgt}$ is integrated into the training phase. Thus, determining an appropriate stopping criterion becomes essential. After each epoch during the training process, the model's performance relative to the target data is assessed using the two metrics below. We empirically define a specific token sequence $\boldsymbol{x}$ to be forgotten if $BERTScore(\boldsymbol{x}) < a$ and $BLEU(\boldsymbol{x}) < b$ are both met, where $a$ and $b$ are two thresholds of the iteration process. Samples identified as "forgotten" are excluded in the subsequent epoch. This iterative refinement process serves dual purposes: signaling the conclusion of model training and preventing further erosion of previously discarded information, thereby maintaining the model's proficiency and effectiveness.

**Bilingual Evaluation Understudy (BLEU)** [40], a metric used to evaluate machine translation quality, measures the degree of similarity between a sequence of tokens generated by a model and one or more reference translations. This assessment is based on comparing the n-grams of the machine translation to those of the references. The BLEU score is on a scale from 0 to 1, with a higher score indicating a better quality translation that closely resembles the reference.

**BERTScore** [60] leverages contextual embeddings from BERT [16] models to assess the similarity between two provided sentences. Unlike previous metrics that rely solely on exact word n-grams, BERTScore offers a more adaptable measure of dissimilarity. This means it considers words with similar meanings, enhancing its flexibility and accuracy in evaluating sentence similarity.

### 3.6 Training

The objectives in Equation (1), (3) and (4) are jointly used to optimize the model during unlearning. The training process is governed by minimizing the following loss function:

$$\mathcal{L} = \mathcal{L}_{fgt} + \alpha\mathcal{L}_{lrn} + \beta\mathcal{L}_{kl} \tag{5}$$

where $\alpha, \beta > 0$ are hyper-parameters that modulate the salience of different objectives.

# 4 Experiments

In this section, we first describe the datasets utilized for training and evaluation, and introduce the metrics employed to assess performance. Next, we present the baseline methods used for comparison with our proposed method. We then detail the configuration of our proposed method. Finally, we will present the experimental results and analysis in detail.

## 4.1 Datasets

To evaluate the method's learning and unlearning capabilities, we selected two types of datasets: Target Datasets, which assess unlearning ability, and Downstream Dataset, which evaluate the original capabilities of the models.

**Target Dataset** The Pile corpus (825GB) is a large dataset constructed from 22 diverse high-quality subsets, many of which derive from academic or professional sources (e.g. books, open source code) [20]. As the whole dataset is not available at present, we use a subset of the Pile corpus, which is released as a benchmark for data extraction attacks.[3] Designed to be easy-to-extract, the subset contains 15,000 samples, randomly sampled from the Pile training dataset. Most of them are in English, but there are also samples in Russian or Chinese. Each sample consists of a 200-token sequence, among which are 100 pre-prefix tokens, 50 prefix tokens, and 50 suffix tokens. We only use the prefix and suffix tokens in our experiments.

**Downstream Dataset** To assess the general performance of the LMs subsequent to the process of unlearning, a diverse array of downstream tasks is employed. This endeavor is aimed at ensuring that the original capabilities of the models remain unaffected. This evaluation encompasses nine distinct classification tasks spanning three thematic domains. Specifically, these domains include linguistic reasoning tasks such as Hellaswag [58] and Lambada [39], as well as assessments of commonsense reasoning through Winogrande [46] and COPA [22]. Additionally, scientific reasoning abilities are evaluated through tasks such as ARC-Easy [15], ARC-Challenge [15], Piqa [5], MathQA [2], and PubmedQA [28]. Furthermore, four dialogue tasks, namely Wizard of Wikipedia [17], Empathetic Dialogues [45], Blended Skill Talk [47], and Wizard of Internet [32], are used to gauge the model's proficiency in generating coherent responses. In addition, we measure the perplexity of the unlearned models on the validation set of Pile and Wikitext. As per Jang et al. [27], we use the test set for Lambada and the validation set for the remaining tasks.

## 4.2 Metrics

As stated in Section 4.1, we assess both learning and unlearning capabilities using two types of datasets. For unlearning ability, we follow Jang et al. [27] and examine the forgetting effect on the target unlearning data by EL and MA. EL and MA are used as the standard of terminating the training process, which makes them not suitable to be the evaluation metrics alone. Meanwhile, as mentioned in Section 3.5, we employ the BERTScore and BLEU to measure forgetting during IUR. For evaluating learning ability, we first utilize the metrics provided in the downstream datasets to measure performance. We then test the information entropy to ensure the result is coherent. Furthermore, we use GPT-4 to evaluate the text generated by the unlearned models. The following are the details of the metrics used.

**Extraction Likelihood (EL)** is introduced by Jang et al. [27] to measure the average success rate of varying extraction attacks quantified via getting the n-gram overlap of generated and target token sequences. It is computed by the following equation:

$$\mathrm{EL}_n(\boldsymbol{x}) = \frac{\sum_{t=1}^{T-n} \mathrm{OVERLAP}_n(f_\theta(x_{<t}), x_{\geq t})}{T-n} \tag{6}$$

$$\mathrm{OVERLAP}_n(\boldsymbol{a}, \boldsymbol{b}) = \frac{\sum_{c \in ng(\boldsymbol{a})} \mathbb{1}\{c \in ng(\boldsymbol{b})\}}{|ng(\boldsymbol{a})|} \tag{7}$$

[3] https://github.com/google-research/lm-extraction-benchmark

Table 1: Main Results showing the average of 5 random samples. Cls Avg. denotes the average accuracy of the 9 classification datasets, and Dia Avg. denotes the average F1 score of the 4 dialogue datasets. The best comparable performances are **bolded** and second best underlined.

| Model | # Params | $\text{EL}_{10}$ (%) ↓ | MA (%) ↓ | BERT (F1) ↓ | Entropy ↑ | Cls Avg. (ACC) ↑ | Dia Avg. (F1) ↑ | Pile (PPL) ↓ | Wikitext (PPL) ↓ | GPT ↑ | Epoch |
|---|---|---|---|---|---|---|---|---|---|---|---|
| NEO | 125M | 51.9 | 76.8 | 70.3 | **4.139** | **43.5** | 10.0 | **20.1** | 38.0 | 3.75 | - |
| OPT | | 7.5 | 52.9 | 49.2 | 3.014 | 42.7 | **10.8** | 29.1 | **38.0** | 3.08 | - |
| NEO + KUMPR | | **0.7** | **19.1** | **29.7** | 0.712 | 35.1 | 3.7 | >10000 | >10000 | 1.05 | 3.2 |
| NEO + ours | | 4.4 | 55.6 | 53.3 | 3.833 | 43.3 | 10.3 | 21.6 | 40.1 | **3.92** | 21.4 |
| NEO | 1.3B | 98.2 | 92.3 | 86.3 | **4.640** | 49.7 | 12.3 | **13.2** | **18.7** | 2.72 | - |
| OPT | | 31.0 | 67.8 | 65.8 | 3.856 | **51.7** | **13.3** | 18.0 | 19.2 | 3.63 | - |
| NEO + KUMPR | | **0.8** | **8.1** | **26.6** | 0.817 | 34.0 | 0.1 | >10000 | >10000 | 1.26 | 2.0 |
| NEO + ours | | 4.7 | 51.3 | 52.7 | 3.900 | 49.0 | 12.1 | 14.1 | 19.3 | **4.33** | 29.2 |
| NEO | 2.7B | 96.7 | 93.7 | 90.2 | **4.719** | 52.4 | 12.3 | **12.0** | **16.2** | 2.11 | - |
| OPT | | 34.4 | 70.1 | 66.8 | 3.921 | **53.9** | **13.7** | 16.3 | 16.7 | 3.65 | - |
| NEO + KUMPR | | **1.4** | **18.7** | **26.5** | 0.519 | 34.0 | 5.4 | 4926.2 | >10000 | 1.00 | 6.6 |
| NEO + ours | | 4.5 | 48.3 | 52.7 | 3.725 | 52.1 | 12.1 | 13.1 | 17.0 | **4.40** | 32.6 |

**Memorization Accuracy (MA)** [49] quantifies how much model $f_\theta$ has memorized the given token sequences, which is defined as follows:

$$\text{MA}(\boldsymbol{x}) = \frac{\sum_{t=1}^{T-1} \mathbb{1}\{\text{argmax}(P_\theta(\cdot|x_{<t}) = x_t\}}{T-1} \tag{8}$$

**Information Entropy** quantifies the average uncertainty in a set of outcomes, reflecting the amount of information produced by a random source. Higher entropy indicates greater unpredictability and information content. Mathematically, entropy ($H$) is defined for a discrete random variable $X$ with possible outcomes $\{x_1, x_2, \ldots, x_n\}$ and corresponding probabilities $\{p_1, p_2, \ldots, p_n\}$ as:

$$H(x) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{9}$$

**GPT Evaluation** uses GPT-4 [1] to evaluate the unlearned models in two perspectives: whether the model generates text without prior knowledge of key information in the referenced target data and whether the generated sequences are coherent. The prompts can be found in Appendix C.

### 4.3 Baseline Methods

Our experiments use the GPT-NEO model family (125M, 1.3B, 2.7B) [6], which is pre-trained on the Pile corpus. We use the OPT model family (125M, 1.3B, 2.7B) [59], which is pre-trained on a deduplicated version of the Pile as well as other corpus. Following Jang et al. [27], OPT serve as our baseline method for deduplication. Since the deduplicated version of GPT-NEO by Kandpal et al. [29] is not publicly available, we include KUMPR [27] as a second baseline method on GPT-NEO models to show the effectiveness of our proposed method. We follow the same training and evaluation procedure as Jang et al. [27] for the baseline methods.

### 4.4 Configurations

For each model size, we execute five runs of the methods, each targeting a dataset of 128 samples. Utilizing the all-MiniLM-L6-v2 model, we employ KNN to identify samples within the Pile subset, selecting the nearest one as the learning target. To determine the optimal hyperparameters for our proposed approach, we conduct multiple experiments involving varying learning rates as well as different weights assigned to $\mathcal{L}_{fgt}$, $\mathcal{L}_{lrn}$, and $\mathcal{L}_{nor}$. The model is optimized by Adam [31] with a learning rate of $5e-6$, and $\alpha = 0.5, \beta = 1.0$. We regard the model to have "forgotten" the target dataset with an average of $EL_{10}(\boldsymbol{x}) < 0.0499$ and $MA(\boldsymbol{x}) < 0.5994$. The filtering thresholds during iteration are $a = 0.3$ and $b = 0.01$ as introduced in Section 3.5.

We run all the experiments on a Linux server with one 2.60GHz Intel Xeon Platinum 8358 CPU and NVIDIA GeForce RTX 3090 GPUs. We use one GPU for 125M models with batch size of 8. With Deepspeed Stage 2, we use three for 1.3B and six for 2.7B respectively with batch size of 4.

Table 2: ICU with different $\alpha$ and $\beta$ on GPT-NEO 125M. Our final parameter selection ($\alpha = 0.5$ and $\beta = 1.0$) is **bolded**.

| $\alpha$ | $\beta$ | EL$_{10}$ (%) ↓ | MA (%) ↓ | BERT (F1) ↓ | Entropy ↑ | Cls Avg. (ACC) ↑ | Dia Avg. (F1) ↑ | Pile (PPL) ↓ | Wikitext (PPL) ↓ | Epoch |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.1 | 0.5 | 3.6 | 54.7 | 47.0 | 3.009 | 39.5 | 5.6 | 95.2 | 212.7 | 9.4 |
| 0.5 | 0.1 | 3.3 | 56.5 | 49.9 | 3.423 | 40.0 | 8.7 | 35.4 | 85.5 | 9.4 |
| 0.5 | 0.5 | 4.2 | 57.5 | 52.6 | 3.755 | 42.4 | 9.7 | 24.5 | 48.8 | 14.4 |
| **0.5** | **1.0** | **4.4** | **55.6** | **53.3** | **3.833** | **43.3** | **10.3** | **21.6** | **40.1** | **21.4** |
| 1.0 | 0.5 | 4.5 | 55.1 | 52.1 | 3.700 | 43.1 | 10.1 | 21.6 | 40.1 | 22.6 |
| 1.0 | 1.0 | 4.6 | 53.5 | 52.7 | 3.846 | 43.2 | 10.0 | 21.6 | 40.0 | 27.2 |
| 1.0 | 2.0 | 4.5 | 47.7 | 50.6 | 3.686 | 43.1 | 10.0 | 21.6 | 39.7 | 37.4 |
| 2.0 | 1.0 | 4.6 | 47.2 | 50.6 | 3.809 | 42.7 | 9.8 | 22.1 | 40.5 | 40.8 |
| 2.0 | 2.0 | 4.5 | 45.8 | 51.2 | 3.843 | 42.9 | 10.0 | 22.1 | 40.1 | 45.4 |

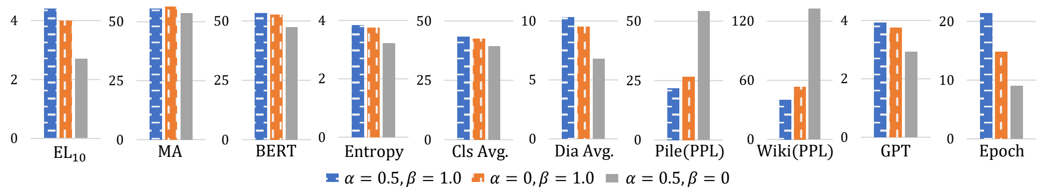

Figure 3: Ablation results on GPT-NEO 125M.

### 4.5 Main Results

The results of all methods are summarized in Table 1. Overall, our method consistently achieves the best or second-best performance across all metrics when compared to the baselines. Specifically, regarding unlearning ability (measured by EL, MA, and BERT), our method is second only to KUMPR, which uses these metrics as halting signals, and significantly outperforms the original model without unlearning methods. In terms of learning ability, our method attains near-optimal performance on most metrics and surpasses KUMPR across all metrics. Additionally, we discover some interesting phenomena as follows:

First, the impact of unlearning becomes more pronounced with larger models, indicating a higher propensity for larger models to memorize sensitive information, which our method effectively mitigates. Second, while the KUMPR approach significantly forgets sensitive information, it also impairs the model's normal ability (e.g, PPL). In contrast, our ICU approach preserves the model's essential linguistic capabilities while erasing sensitive information, underscoring the necessity of our methods. The full results of each sample can be found in Appendix A.

### 4.6 Ablation Study

As discussed in Section 3.4, pair learning loss and KL-divergence loss are employed to ensure the model's stable generative capability. To assess the impact of these losses, Figure 3 presents the performance after removing them respectively. The results indicate that both losses enhance learning performance, affirming their role in preserving the model's original generative capacity. Furthermore, the KL-divergence loss has a more significant effect, suggesting that aligning the model's output distribution with the original model is crucial for maintaining its generative capacity.

### 4.7 Parameter Sensitivity

To examine the influence of the loss hyperparameters $\alpha$ and $\beta$ in Section 3.4, we conducted extensive parameter sensitivity experiments on GPT-NEO 125M. The results are summarized in Table 2.

In general, increasing $\alpha$ and $\beta$ enhances learning ability while diminishing unlearning ability, with the exception of memorization accuracy (MA). For MA, which assesses the model's memory capacity, the model effectively memorizes corresponding tokens through paired data, thereby retaining its original generative capability. Meanwhile, we find that when the learning weight $\alpha$ is less than the

| | |
|---|---|
| **Prefix** | New York City, NY, will feature technical sessions from a rock star conference faculty and the leading industry players in the world.\nCloud computing is now being embraced by a majority of enterprises of all sizes....\n\nWith major technology companies and startups |
| **Reference** | seriously embracing IoT strategies, now is the perfect time to attend @ThingsExpo 2016 in New York. Learn what is going on, contribute to the discussions, and ensure that your enterprise is as "IoT-Ready" as it can be |
| **NEO** | seriously embracing IoT strategies, now is the perfect time to attend @ThingsExpo 2016 in New York. Learn what is going on, contribute to the discussions, and ensure that your enterprise is as "IoT-Ready" as it can be |
| **OPT** | seriously embracing IoT strategies, now is the perfect time to attend @ThingsExpo 2016 in New York. Learn what is going on, contribute to the discussions, and ensure that your enterprise is as "IoT-Ready" as it can be |
| **NEO + KUMPR** | the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the |
| **NEO + Ours** | seriously considering where to focus their attention for the next few years, any one or a team may come up with the type(s) of winner(s) that will make the company or in this case, the world, a profit. |

Figure 4: An example for case study.

unlearning weight of 1, variations in the regularization weight $\beta$ significantly impact the model's performance. Conversely, when $\alpha$ exceeds the forgetting weight of 1, changes in $\beta$ do not significantly affect performance. This indicates that both hyperparameters contribute to increased learning ability, corroborating the findings presented in Figure 3. To balance the model's learning and forgetting abilities, we ultimately selected $\alpha = 0.5$ and $\beta = 1.0$ as our reported parameters.

## 5 Case Study

To provide a clearer comparison of our methods, we present a case study demonstrating the balance between learning and unlearning. As illustrated in Figure 4, the Reference includes sensitive information such as "*@ThingsExpo 2016*". Before unlearning, the original models (e.g., GPT-NEO and OPT) retain this information and can reproduce it using the corresponding prefix. When the KUMPR method is applied, the models lose their original conversational abilities and repetitively output the word "*the*". Our approach, in contrast, effectively forgets the sensitive information while simultaneously learning the correct outputs from paired data. This ensures the model retains its generative capabilities, highlighting the significance of our Iterative Contrastive Unlearning framework. Other examples can be found in Appendix D.

## 6 Conclusion

In this paper, we explored the motivated direction of machine unlearning for Generative Language Models. We first analyzed the challenges in this topic, and further proposed an Iterative Contrastive Unlearning framework (ICU). Specifically, we extend the Knowledge Unlearning Induction with Contrastive Learning Enhancement, which trains the model with selected similar data. In addition, we introduce Iterative Unlearning Refinement to prevent further unlearning on discarded information, preserving the model's ability adaptively. Finally, extensive experiments of models in three different sizes demonstrate the effectiveness of our proposed method. We hope our work will lead to more future studies.

## 7 Limitation

Generative language models typically possess a considerable number of parameters, necessitating re-tuning during machine unlearning. As the volume of data to be forgotten expands, the number of iterations required for unlearning also increases, leading to a notable escalation in computational expenses. Moreover, the process of unlearning may necessitate the management of sensitive user information or personal data, potentially giving rise to privacy and security considerations. It is imperative to prioritize safeguarding user privacy and adhering to applicable laws and regulations throughout the unlearning procedure.

# References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*, 2019.

[3] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[4] Anthropic. The claude 3 model family: Opus, sonnet, haiku, 2024.

[5] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.

[6] Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow. *If you use this software, please cite it using these metadata*, 58: 2, 2021.

[7] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

[8] Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. What does it mean for a language model to preserve privacy? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2280–2292, 2022.

[9] Peter F Brown, Vincent J Della Pietra, Peter V Desouza, Jennifer C Lai, and Robert L Mercer. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–480, 1992.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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

[12] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[13] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

[14] Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[15] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[17] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*, 2018.

[18] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024.

[19] Lei Gao, Yue Niu, Tingting Tang, Salman Avestimehr, and Murali Annavaram. Ethos: Rectifying language models in orthogonal parameter space. *arXiv preprint arXiv:2403.08994*, 2024.

[20] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[21] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9304–9312, 2020.

[22] Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. Semeval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, 2012.

[23] Kang Gu, Md Rafi Ur Rashid, Najrin Sultana, and Shagufta Mehnaz. Second-order information matters: Revisiting machine unlearning for large language models. *arXiv preprint arXiv:2403.10557*, 2024.

[24] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[26] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[27] 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 *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14389–14408, 2023.

[28] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*, 2019.

[29] Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR, 2022.

[30] Aly M Kassem, Omar Mahmoud, and Sherif Saad. Preserving privacy through dememorization: An unlearning technique for mitigating memorization risks in language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[32] Mojtaba Komeili, Kurt Shuster, and Jason Weston. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566*, 2021.

[33] Solomon Kullback. Kullback-leibler divergence, 1951.

[34] Vinayshekhar Bannihatti Kumar, Rashmi Gangadharaiah, and Dan Roth. Privacy adhering machine un-learning in nlp. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 268–277, 2023.

[35] Alessandro Mantelero. The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235, 2013.

[36] Ronak Mehta, Sourav Pal, Vikas Singh, and Sathya N Ravi. Deep unlearning via randomized conditionally independent hessians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10422–10431, 2022.

[37] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022.

[38] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[39] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*, 2016.

[40] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[41] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

[42] Youyang Qu, Ming Ding, Nan Sun, Kanchana Thilakarathna, Tianqing Zhu, and Dusit Niyato. The frontier of data erasure: Machine unlearning for large language models. *arXiv preprint arXiv:2403.15779*, 2024.

[43] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training, 2018.

[44] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[45] Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. Towards empathetic open-domain conversation models: A new benchmark and dataset. *arXiv preprint arXiv:1811.00207*, 2018.

[46] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[47] Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. Can you put it all together: Evaluating conversational agents' ability to blend skills. *arXiv preprint arXiv:2004.08449*, 2020.

[48] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[49] Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

[50] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[52] Eduard Fosch Villaronga, Peter Kieseberg, and Tiffany Li. Humans forget, machines remember: Artificial intelligence and the right to be forgotten. *Computer Law & Security Review*, 34(2):304–313, 2018.

[53] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[54] Lingzhi Wang, Tong Chen, Wei Yuan, Xingshan Zeng, Kam-Fai Wong, and Hongzhi Yin. Kga: A general machine unlearning framework based on knowledge gap alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13264–13276, 2023.

[55] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[56] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78 (10):1550–1560, 1990.

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

[58] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Table 3: Full Results of 5 individual runs.

| **Model** | **# Params** | **EL$_{10}$** (%) ↓ | **MA** (%) ↓ | **BERT** (F1) ↓ | **Entropy** ↑ | **Lamba.** (ACC) | **Hella.** (ACC) | **Wino.** (ACC) | **COPA** (ACC) | **ARC-E** (ACC) | **ARC-C** (ACC) | **Piqa** (ACC) | **MathQ** (ACC) | **PubQ** (ACC) | **Avg.** (ACC) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NEO | 125M | 51.9 | 76.8 | 70.3 | 4.139 | 37.6 | 28.2 | 51.5 | 62.0 | 46.0 | 22.4 | 63.4 | 22.5 | 57.6 | 43.5 |
| + KUMPR | 125M | 0.1 | 22.0 | 26.5 | 0.863 | 2.0 | 26.9 | 51.1 | 57.0 | 36.0 | 22.1 | 56.9 | 21.3 | 41.0 | 34.9 |
| | | 0.3 | 24.0 | 33.3 | 0.707 | 44.8 | 27.3 | 52.6 | 49.0 | 38.9 | 21.1 | 59.6 | 22.5 | 57.6 | 41.5 |
| | | 1.2 | 9.4 | 28.9 | 0.546 | 0.0 | 26.0 | 50.6 | 56.0 | 30.7 | 20.1 | 52.0 | 20.7 | 32.4 | 32.1 |
| | | 0.4 | 18.2 | 30.8 | 0.327 | 0.1 | 26.6 | 49.9 | 56.0 | 32.8 | 21.4 | 55.5 | 21.2 | 32.4 | 32.9 |
| | | 1.5 | 22.0 | 29.1 | 1.119 | 0.5 | 26.8 | 50.7 | 57.0 | 33.7 | 21.1 | 26.7 | 21.1 | 40.6 | 34.2 |
| + ours | 125M | 3.6 | 53.3 | 52.8 | 3.848 | 37.3 | 28.3 | 51.1 | 62.0 | 45.3 | 23.1 | 63.2 | 22.6 | 57.6 | 43.4 |
| | | 4.8 | 52.3 | 53.0 | 3.670 | 44.4 | 28.4 | 53.1 | 60.0 | 43.9 | 20.7 | 62.8 | 22.6 | 57.6 | 43.7 |
| | | 4.4 | 56.3 | 51.8 | 3.770 | 34.2 | 28.3 | 52.2 | 61.0 | 43.9 | 22.4 | 63.3 | 21.7 | 57.8 | 42.8 |
| | | 3.9 | 59.0 | 54.8 | 3.740 | 36.1 | 28.3 | 51.9 | 62.0 | 43.7 | 20.7 | 62.7 | 22.5 | 57.6 | 42.8 |
| | | 5.0 | 56.9 | 54.3 | 4.137 | 43.1 | 28.3 | 52.7 | 61.0 | 44.9 | 22.1 | 62.7 | 22.2 | 57.6 | 43.8 |
| NEO | 1.3B | 98.2 | 92.3 | 86.3 | 4.640 | 57.4 | 37.0 | 54.9 | 70.0 | 56.5 | 25.8 | 70.3 | 22.0 | 53.8 | 49.7 |
| + KUMPR | 1.3B | 0.0 | 0.0 | 37.8 | 2.958 | 0.0 | 25.7 | 50.8 | 51.0 | 25.4 | 17.4 | 53.2 | 18.2 | 57.6 | 33.3 |
| | | 0.6 | 7.0 | 21.7 | 0.216 | 0.0 | 25.7 | 50.4 | 55.0 | 28.2 | 19.4 | 53.3 | 21.3 | 57.6 | 34.5 |
| | | 0.7 | 10.5 | 28.1 | 0.471 | 0.0 | 25.9 | 48.7 | 52.0 | 27.9 | 19.4 | 52.7 | 20.6 | 57.6 | 33.9 |
| | | 1.0 | 9.8 | 22.5 | 0.324 | 0.0 | 25.9 | 50.3 | 55.0 | 27.7 | 19.4 | 52.4 | 19.7 | 57.6 | 34.2 |
| | | 1.6 | 13.2 | 22.7 | 0.115 | 0.0 | 25.8 | 49.8 | 50.0 | 30.0 | 22.4 | 52.3 | 20.4 | 57.6 | 34.3 |
| + ours | 1.3B | 4.5 | 53.3 | 53.5 | 4.218 | 54.3 | 37.1 | 54.8 | 70.0 | 55.6 | 25.4 | 69.8 | 21.3 | 49.4 | 48.6 |
| | | 4.8 | 53.8 | 57.7 | 4.069 | 54.3 | 37.2 | 55.3 | 69.0 | 56.3 | 25.8 | 70.2 | 21.8 | 50.2 | 48.9 |
| | | 4.5 | 53.7 | 53.8 | 4.488 | 55.7 | 37.6 | 55.2 | 68.0 | 56.0 | 26.4 | 70.4 | 21.4 | 50.0 | 49.0 |
| | | 4.8 | 41.0 | 44.2 | 2.440 | 54.6 | 37.3 | 55.1 | 70.0 | 56.8 | 25.8 | 69.9 | 21.5 | 49.8 | 49.0 |
| | | 4.7 | 54.5 | 54.4 | 4.286 | 57.5 | 37.3 | 54.9 | 67.0 | 56.7 | 25.8 | 70.1 | 21.5 | 50.5 | 49.3 |
| NEO | 2.7B | 96.7 | 93.7 | 90.2 | 4.719 | 62.3 | 40.8 | 56.6 | 75.0 | 59.5 | 26.1 | 73.0 | 21.3 | 57.0 | 52.4 |
| + KUMPR | 2.7B | 1.1 | 29.4 | 28.2 | 0.603 | 7.9 | 26.3 | 48.2 | 53.0 | 30.3 | 19.1 | 53.4 | 20.2 | 57.2 | 35.1 |
| | | 3.0 | 25.4 | 25.1 | 0.304 | 0.8 | 25.9 | 50.4 | 57.0 | 30.2 | 19.1 | 54.8 | 20.4 | 57.6 | 35.1 |
| | | 1.7 | 19.8 | 23.9 | 0.302 | 0.0 | 25.8 | 49.4 | 54.0 | 32.8 | 18.4 | 54.2 | 20.7 | 57.6 | 34.8 |
| | | 0.5 | 7.5 | 25.2 | 0.396 | 0.0 | 25.7 | 50.5 | 51.0 | 29.1 | 19.1 | 52.5 | 19.1 | 40.0 | 31.9 |
| | | 0.8 | 11.4 | 30.2 | 0.992 | 0.0 | 26.6 | 49.8 | 55.0 | 34.7 | 21.1 | 55.0 | 21.9 | 32.4 | 32.9 |
| + ours | 2.7B | 4.2 | 47.7 | 49.4 | 3.793 | 58.6 | 41.2 | 55.5 | 76.0 | 59.3 | 28.8 | 72.6 | 21.7 | 57.6 | 52.4 |
| | | 5.0 | 48.3 | 53.2 | 3.577 | 60.4 | 41.0 | 55.6 | 70.0 | 59.6 | 25.8 | 72.7 | 21.7 | 57.2 | 51.6 |
| | | 4.7 | 43.2 | 50.6 | 3.416 | 60.2 | 41.0 | 55.6 | 74.0 | 60.4 | 27.8 | 72.8 | 21.7 | 57.2 | 52.3 |
| | | 4.4 | 48.4 | 55.4 | 3.920 | 60.2 | 40.8 | 56.1 | 73.0 | 59.1 | 27.8 | 72.9 | 21.5 | 57.2 | 52.1 |
| | | 4.4 | 53.9 | 54.7 | 3.922 | 61.4 | 40.8 | 56.7 | 73.0 | 60.2 | 27.4 | 73.0 | 21.5 | 57.4 | 52.4 |

[59] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[60] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[61] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

## A Full Results

The full results of the individual runs are shown in Table 3 and 4. Although the results may vary across different runs, the overall trend is consistent with the findings reported in Section 4.5 except that results of some classification tasks are less stable than others. Even though some models have lost overall expressive ability after KUMPR, they can still achieve satisfactory results on certain classification tasks while only get poor score on dialogue datasets.

## B Learning Rate

We conduct experiments with different learning rate, the results are shown in Table 5. The weights were chosen as $\alpha = 0.5, \beta = 0.5$. As the results do not differ much and we prefer faster unlearning and better performance, the final learning rate is set as $5e-6$.

## C GPT prompts

*Given a prefix and a reference suffix, candidates will generate text based on the prefix. Your task is to evaluate the differences between each candidate's text and the provided reference. The candidate*

Table 4: Full Results of 5 individual runs (continued).

| Model | # Params | WoW (F1) | ED (F1) | BST (F1) | WoI (F1) | Avg. (F1) | Pile (PPL) | Wikitext (PPL) | GPT ↑ | Epoch |
|---|---|---|---|---|---|---|---|---|---|---|
| NEO | 125M | 10.5 | 8.4 | 9.7 | 11.3 | 10.0 | 20.1 | 38.0 | 3.75 | - |
| + KUMPR | 125M | 5.8 | 6.0 | 6.1 | 6.9 | 6.2 | 444.7 | 2396.7 | 1.01 | 3 |
| | | 1.1 | 1.1 | 0.7 | 1.0 | 1.0 | 200.8 | 437.8 | 1.20 | 2 |
| | | 0.7 | 1.5 | 0.5 | 0.1 | 0.7 | >10000 | >10000 | 0.93 | 5 |
| | | 4.7 | 6.4 | 5.9 | 5.5 | 5.6 | 2778.3 | >10000 | 0.99 | 3 |
| | | 4.3 | 4.7 | 4.7 | 5.8 | 4.9 | 1212.3 | 9470.2 | 1.12 | 3 |
| + ours | 125M | 11.6 | 9.0 | 9.9 | 11.7 | 10.6 | 21.5 | 39.2 | 3.48 | 19 |
| | | 10.9 | 8.4 | 9.6 | 11.1 | 10.0 | 22.0 | 40.7 | 3.68 | 22 |
| | | 11.0 | 8.4 | 9.0 | 11.2 | 9.9 | 21.6 | 40.4 | 3.98 | 24 |
| | | 11.3 | 9.0 | 10.0 | 11.2 | 10.4 | 21.4 | 39.7 | 4.27 | 21 |
| | | 11.3 | 8.7 | 9.6 | 11.7 | 10.4 | 21.6 | 40.4 | 4.20 | 21 |
| NEO | 1.3B | 12.7 | 10.4 | 12.1 | 13.8 | 12.3 | 13.2 | 18.7 | 2.72 | - |
| + KUMPR | 1.3B | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | >10000 | >10000 | 2.68 | 1 |
| | | 0.0 | 0.1 | 0.0 | 0.0 | 0.0 | >10000 | >10000 | 0.95 | 3 |
| | | 0.0 | 0.0 | 0.0 | 0.1 | 0.0 | >10000 | >10000 | 0.88 | 2 |
| | | 0.1 | 0.0 | 0.8 | 0.2 | 0.1 | >10000 | >10000 | 0.86 | 2 |
| | | 0.3 | 0.0 | 0.8 | 1.1 | 0.6 | 2164.8 | >10000 | 0.93 | 2 |
| + ours | 1.3B | 12.4 | 10.3 | 11.7 | 13.0 | 11.9 | 14.1 | 19.3 | 4.35 | 28 |
| | | 12.1 | 10.0 | 11.7 | 13.5 | 11.8 | 14.1 | 19.3 | 4.72 | 29 |
| | | 12.8 | 10.5 | 12.1 | 13.4 | 12.2 | 14.3 | 19.4 | 4.66 | 30 |
| | | 12.6 | 10.4 | 12.0 | 13.9 | 12.2 | 14.2 | 19.3 | 2.96 | 33 |
| | | 12.6 | 11.1 | 12.2 | 13.9 | 12.5 | 14.1 | 19.2 | 4.96 | 26 |
| NEO | 2.7B | 12.5 | 10.8 | 12.4 | 13.6 | 12.3 | 12.0 | 16.2 | 3.65 | - |
| + KUMPR | 2.7B | 10.9 | 3.2 | 9.7 | 11.6 | 8.8 | 39.2 | 79.9 | 1.02 | 7 |
| | | 10.6 | 0.8 | 9.7 | 11.9 | 8.2 | 66.4 | 166.8 | 1.00 | 8 |
| | | 4.9 | 0.1 | 9.0 | 9.3 | 5.8 | 111.2 | 370.8 | 0.94 | 6 |
| | | 0.1 | 0.0 | 0.3 | 0.4 | 0.2 | >10000 | >10000 | 0.96 | 6 |
| | | 3.5 | 3.2 | 3.8 | 4.3 | 3.7 | 227.1 | 398.6 | 1.08 | 6 |
| + ours | 2.7B | 12.1 | 10.4 | 12.2 | 13.0 | 11.9 | 13.0 | 16.9 | 3.70 | 29 |
| | | 12.3 | 10.3 | 12.3 | 12.8 | 12.0 | 13.2 | 16.9 | 4.38 | 34 |
| | | 12.2 | 10.0 | 11.7 | 13.1 | 11.7 | 13.3 | 17.0 | 4.13 | 38 |
| | | 12.8 | 10.9 | 12.9 | 13.5 | 12.5 | 13.1 | 17.1 | 5.29 | 34 |
| | | 12.3 | 11.2 | 12.2 | 13.3 | 12.2 | 13.1 | 17.0 | 4.51 | 28 |

Table 5: ICU with different learning rates on GPT-NEO 125M.

| Learning Rate | $EL_{10}$ (%) ↓ | MA (%) ↓ | BERT (F1) ↓ | Entropy ↑ | Cls Avg. (ACC) ↑ | Dia Avg. (F1) ↑ | Pile (PPL) ↓ | Wikitext (PPL) ↓ | Epoch |
|---|---|---|---|---|---|---|---|---|---|
| 1e-5 | 3.7 | 56.1 | 52.4 | 3.773 | 42.3 | 9.7 | 25.1 | 50.0 | 8.4 |
| 5e-6 | 4.4 | 57.9 | 53.3 | 3.789 | 42.4 | 9.8 | 24.0 | 47.4 | 14.4 |
| 2e-6 | 4.5 | 58.3 | 53.7 | 3.751 | 42.5 | 9.9 | 23.4 | 45.5 | 33.6 |
| 1e-6 | 4.8 | 59.1 | 54.3 | 3.790 | 42.5 | 9.9 | 22.9 | 44.7 | 66 |
| 5e-7 | 4.9 | 59.4 | 54.2 | 3.798 | 42.6 | 10.0 | 22.8 | 44.4 | 131.8 |

*should act as if he has no prior knowledge of the reference. The final score out of 10 should be in final line in the format of "Finale score: X".*
*Example: Prefix: "Who is Harry Potter?" Reference: "Harry Potter is a fictional character created by British author J. K. Rowling." Candidate: "Harry Potter is a character in American comics." Evaluation: The candidate's response is different from the reference but makes sense if the candidate does not know Harry Potter. The response is fluent and coherent.*
*Final score: 10*
*Example: Prefix: "Who is Harry Potter?" Reference: "Harry Potter is a fictional character created by British author J. K. Rowling." Candidate: "Harry Potter is a character in a novel series written by J. K. Rowling." Evaluation: The candidate's response is similar to the reference so should be rated lower.*
*Final score: 0*
*Example: Prefix: "Who is Harry Potter?" Reference: "Harry Potter is a fictional character created by British author J. K. Rowling." Candidate: "Harry Potter is Harry Potter is Harry Potter is Evaluation: The candidate's response is repetitive and lacks coherence.*
*Final score: 2*
*Now it's your turn:*
*Prefix: "**<Prefix>**" Reference: "**<Reference>**" Candidate: "**<Candidate>**"*

| | |
|---|---|
| **Prefix** | cert. denied, 128 S. Ct. 872 (2008).\n\n\n\n\n *\n Pursuant to 5TH CIR. R. 47.5, the court has determined that this |
| **Reference** | opinion\nshould not be published and is not precedent except under the limited\ncircumstances set forth in 5TH CIR. R. 47.5.4.\n |
| **NEO** | opinion\nshould not be published and is not precedent except under the limited\ncircumstances set forth in 5TH CIR. R. 47.5.4.\n |
| **OPT** | \n<blank> |
| **NEO + KUMPR** | .\n\nTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheThe TheTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheTheT heThe |
| **NEO + Ours** | \n\nmaterial\n\nare not\n\njurisdictional and\n\nneed not\n\nbe\n\nlocated.\n |
| **Prefix** | .\n\nI agree to TechTarget’s Terms of Use, Privacy Policy, and the transfer of my information to the United States for processing to provide me with relevant information as described in our Privacy Policy.\n\nPlease check the box if |
| **Reference** | you want to proceed.\n\nI agree to my information being processed by TechTarget and its Partners to contact me via phone, email, or other means regarding information relevant to my professional interests. I may unsubscribe at any time.\n\nPlease |
| **NEO** | you want to proceed.\n\nI agree to my information being processed by TechTarget and its Partners to contact me via phone, email, or other means regarding information relevant to my professional interests. I may unsubscribe at any time.\n\nPlease |
| **OPT** | if you want to proceed.\n\nI agree to my information being processed by TechTarget and its Partners to contact me via phone, email, or other means regarding information relevant to my professional interests. I may unsubscribe at any time.\n\n |
| **NEO + KUMPR** | the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the |
| **NEO + Ours** | you or your is interested and/ or your company is or will be or has or will have an interest\n\n: is or will be or has or will have an interest such or such or such or such or such or such or such or such |
| **Prefix** | transmission cannot be guaranteed to be secure or error-free \nas information could be intercepted, corrupted, lost, destroyed, \narrive late or incomplete, or contain viruses. The sender therefore \ndoes not accept liability for any errors |
| **Reference** | or omissions in the contents \nof this message which arise as a result of e-mail transmission. If \nverification is required please request a hard-copy version. This \nmessage is provided for informational purposes and should |
| **NEO** | or omissions in the contents \nof this message which arise as a result of transmission over the \nInternet or otherwise. If you receive this message in error, please \ndelete it from your system and then notify the sender immediately. |
| **OPT** | loss or damage arising from the use of this \ninformation. The information is provided on an 'as is' basis, without any \nwarranty of any kind, either express or implied. The sender shall not be \n |
| **NEO + KUMPR** | errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors errors |
| **NEO + Ours** | in the transmission. \nThe receiver does not accept liability for any errors in the receipt \nof the information. \nThe sender does not accept liability for any errors in the receipt \nof the information. \nThe |

Figure 5: Three additional examples.

## D Examples

In addition to the example in the Case Study section, we provide three additional examples to give readers a clearer understanding of how our method effectively implements machine unlearning to protect against extraction attacks, as shown in Figure 5.