Protecting Privacy Through Approximating Optimal Parameters for Sequence Unlearning in Language Models

Dohyun Lee[1*] **Daniel Rim**[2*] **Minseok Choi**[1] **Jaegul Choo**[1]
[1]KAIST AI, [2]Hyundai Motor Company
{aiclaudev, minseok.choi, jchoo}@kaist.ac.kr
{drim}@hyundai.com

Abstract

Although language models (LMs) demonstrate exceptional capabilities on various tasks, they are potentially vulnerable to extraction attacks, which represent a significant privacy risk. To mitigate the privacy concerns of LMs, machine unlearning has emerged as an important research area, which is utilized to induce the LM to selectively forget about some of its training data. While completely retraining the model will guarantee successful unlearning and privacy assurance, it is impractical for LMs, as it would be time-consuming and resource-intensive. Prior works efficiently unlearn the target token sequences, but upon subsequent iterations, the LM displays significant degradation in performance. In this work, we propose **P**rivacy Protection via **O**ptimal **P**arameters (POP), a novel unlearning method that effectively forgets the target token sequences from the pretrained LM by applying optimal gradient updates to the parameters. Inspired by the gradient derivation of complete retraining, we approximate the optimal training objective that successfully unlearns the target sequence while retaining the knowledge from the rest of the training data. Experimental results demonstrate that POP exhibits remarkable retention performance post-unlearning across 9 classification and 4 dialogue benchmarks, outperforming the state-of-the-art by a large margin. Furthermore, we introduce Remnant Memorization Accuracy that quantifies privacy risks based on token likelihood and validate its effectiveness through both qualitative and quantitative analyses.

1 Introduction

Language models (LMs) pretrained on a substantial amount of text have demonstrated remarkable performance on various tasks. One of the most important factors in improving performance is training on larger datasets, often containing more than trillions of tokens in the latest models. The datasets used to train such models, however, inevitably contain private information, as it is impossible to check all tokens for privacy concerns. Machine learning models are well-known for being vulnerable to manipulations that can expose the training data, potentially generating exact strings from the training data (Carlini et al., 2019, 2021). Additionally, it has been reported that extracting exact training data becomes easier as models scale to larger sizes (Carlini et al., 2022). With many LMs publicly available (Zhao et al., 2023), the importance of managing the inherent privacy risks in such models has also increased. Moreover, all practitioners are required to delete personal information from machine learning models when requested, to comply with the "Right To Be Forgotten (RTBF)" (Hoofnagle

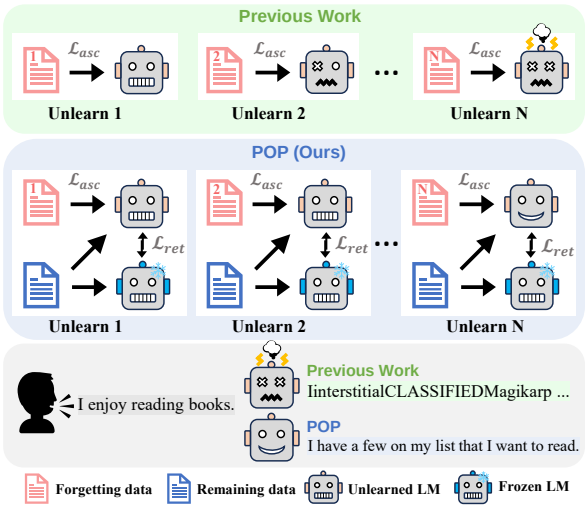


Figure 1: **Our proposed method.** $\mathcal{L}_{asc}$ is the gradient ascent loss for unlearning the target data. If utilized alone, significant performance degradation occurs. By applying both retain loss $\mathcal{L}_{ret}$ and $\mathcal{L}_{asc}$, our method unlearns the target data *and* retains the LM performance. For example, after applying unlearning in succession, previous work demonstrates catastrophic degradation, while POP demonstrates successful retention. Our approach is detailed in Section 3.

[*] Equal contribution

Findings of the Association for Computational Linguistics: ACL 2024, pages 15820–15839
August 11-16, 2024

et al., 2019) from the European Union's General Data Protection Regulation agreement (Voigt and Von dem Bussche, 2017) and the United States California Consumer Privacy Act (Pardau, 2018). To mitigate the potential data leakage and comply with privacy regulations, machine unlearning has emerged as an important research area.

Previous machine unlearning approaches attempted to achieve exact unlearning by removing all private information from the training data, or designed algorithms to ensure differential privacy (DP) (Anil et al., 2022). Some have proposed changes to the training process to make unlearning easier for the pretrained model (Thudi et al., 2022). These efforts require re-training of LMs every time an individual practices one's RTBF, which is extremely expensive and time-consuming. Although the complete re-training of LMs would be optimal for machine unlearning, the cost of doing so is too severe, making such approaches impractical. Others proposed approximate unlearning of target token sequence, applying only a few parameter updates to the pretrained LMs (Jang et al., 2023), or utilizing reinforcement learning feedback loop via proximal policy optimization to unlearn the token sequences (Kassem et al., 2023). Jang et al. (2023) assert that a simple gradient ascent on the target token sequences can be effective at forgetting them. This method is not optimal, as gradient ascent only applies a *portion* of the optimal gradient updates to the parameters. As shown in Fig. 1, adherence to multiple unlearning requests results in accumulation of errors from inadequate approximations, ultimately accumulating to a significant amount. While it may successfully unlearn a few instances in a single batch, the degradation in performance will make the LM useless after unlearning multiple sequences. As ensuring the retention of LM performance is just as important as unlearning the target token sequences, any method that cannot guarantee unlearning *and* retention, after multiple requests, is not a viable machine unlearning solution. Kassem et al. (2023) demonstrated better retention of language model capabilities in various NLP benchmarks, but their method requires all token sequences that come before the target token sequence in the training data to unlearn the target token sequence. As there can be multiple token sequences that come before a target token sequence, their method is extremely difficult to apply in real-world applications.

In this paper, we propose **P**rivacy Protection via **O**ptimal **P**arameters (POP), which applies the optimal gradient updates for sequence unlearning. The gold standard for machine unlearning is a complete retraining from scratch, after removing the target token sequences from the training data. Without committing excessive approximations, POP attempts to emulate the gold standard, updating the parameters as if they were never trained on the target token sequence. After carefully examining the overall gradient updates of the training process, we identify the optimal parameter updates for machine unlearning. Based on our findings, we formalize our solution, which utilizes the pretrained weights, the target token sequence, and the remaining data to achieve inexpensive and optimal machine unlearning. As shown in Fig. 1, POP successfully unlearns the target sequence and ensures the retention of general LM performance post-unlearning, even in a sequential unlearning context where the model applies unlearning requests in succession. Moreover, POP does not require any token prefixes from the training data to unlearn token sequences, rendering it a more viable choice in real-world settings.

We also present Remnant Memorization Accuracy (RMA), a novel metric for quantifying privacy risks. Compared to other sequence unlearning metrics, RMA is the most strict and provides the most robust privacy protection, as it considers the *probabilities* of tokens within the target sequences. When utilized in an unlearning context, RMA can be used as a guideline to determine when unlearning is completed. As it would be unnecessary to excessively unlearn the target sequence from the model, setting an appropriate threshold for unlearning is important. We perform experiments by setting empirical thresholds for each unlearning metric and demonstrate RMA's superiority in providing the strongest privacy protection.

Overall, our contributions are threefold:

- We present POP, a robust knowledge unlearning method that successfully unlearns a target sequence while retaining the general performance of the LM.
- We demonstrate POP's superior performance in both the batch and sequential unlearning processes through quantitative and qualitative analyses.
- We propose RMA, a novel metric for quantifying privacy risks, and demonstrate its strength in providing robust privacy guarantees.

2 Related Work

Data Preprocessing This approach aims to achieve exact unlearning by removing the target sequences from training data through preprocessing methods. This can effectively mitigate privacy risks for sequences that follow easily identifiable formats, such as phone numbers, email addresses, and more (Aura et al., 2006; Dernoncourt et al., 2016; Lison et al., 2021). Private information, however, is context-dependent (Brown et al., 2022), making it impossible to completely remove all private data. Another method that is applied prior to training is data deduplication (Kandpal et al., 2022), which showed improved robustness against data extraction attacks by removing duplicate data from the pretraining corpus. Although this may be effective at mitigating overall privacy risks, it cannot be utilized in a targeted manner for unlearning a specific target token sequence.

Differential Privacy DP preserving methods look to prevent memorization of individual training examples (Dwork et al., 2006; Dwork, 2006; Abadi et al., 2016). Although such methods have been effective in fine-tuning LMs (Yu et al., 2021; Li et al., 2021), pretraining LMs with DP significantly reduces performance, requires expensive computations, and converges very slowly (Anil et al., 2022). Furthermore, as it is impossible to define privacy boundaries for natural language (Brown et al., 2022), DP methods are inherently not applicable for target sequence unlearning.

Knowledge Editing Knowledge editing methods modify LMs to achieve a diverse set of objectives. Some apply various transformations to the neural representations to identify and remove specific concepts (Ravfogel et al., 2022b,a; Belrose et al., 2023). Some apply other methods to maintain the relevancy of the LMs, efficiently updating the underlying knowledge without degrading their performance (Yao et al., 2023). Although these methods alter the pretrained LM for their respective goals, none are designed for the task of unlearning specific token sequences.

Sequence Unlearning For unlearning specific token sequences, Jang et al. (2023) proposed a simple gradient-based solution in reducing the generation likelihood of forgetting token sequences. Although the proposed solution can approximately remove a target token sequence, it also suffers from a large degradation in overall language modeling performance. This downside is even more evident when unlearning multiple sequences in succession, making it impractical for real-world use. Our method not only effectively trains the LMs to forget the target sequence, but also mitigates the potential problems from approximation of the gradients.

More recently, Kassem et al. (2023) presented DeMem, which utilizes a reinforcement learning feedback loop via proximal policy optimization to unlearn token sequences that follow the given prefix sequences. Although DeMem achieves sequence unlearning, it is fundamentally different from ours as their goal is to mitigate memorization by altering the token sequences that follow the given prefix sequences. In a real-world setting with multiple RTBF requests, however, defining the correct set of prefixes for a target token sequence will be difficult, and missing a prefix could present privacy concerns. An ideal unlearning solution should remove token sequences without relying on identifying all possible prefix sequences. POP provide a more robust unlearning solution, by eliminating the generation likelihood of any token sequences.

3 Methodology

3.1 Problem Definition

Given i-th sequence of tokens $\mathbf{x}_i = (x_1, \ldots, x_T)$ in the pretraining dataset $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, causal language modeling minimizes the negative log-likelihood loss:

$$\mathcal{L}(\mathbf{x}_i; \theta) = -\sum_{t=1}^{T} \log(p_\theta(x_t | x_{<t})). \quad (1)$$

Assuming that the update occurred for each sequence, and without considering the learning rate, we define the update step as

$$\theta_j = \theta_{j-1} - \nabla_\theta \mathcal{L}(\mathbf{x}_j; \theta_{j-1}), \quad (2)$$

where θ_j denotes the parameters which is updated for each sequence on $\{\mathbf{x}_1, \ldots, \mathbf{x}_j\}$. Notably, the pretrained model θ_{ptr} is equal to θ_N, as both are trained on N token sequences. Subsequently, our unlearning objective is to approximate the optimal parameters achievable from complete retraining, i.e., θ_{rtr}, from the pretrained model θ_{ptr}. Concretely, θ_{ptr} refers to the parameters before unlearning the target sequence $\mathbf{x}^F \in \mathcal{D}^F$, where $\mathcal{D}^F \subset \mathcal{D}$ contains the target sequence, and θ_{rtr} denotes the optimal parameters obtained from retraining on the remaining data $\mathcal{D}^R = \mathcal{D} \setminus \mathcal{D}^F$.

3.2 POP

In this section, we elaborate on the details of POP and its derivations for the optimal parameter updates for sequence unlearning.

Approximation of θ_{rtr} Suppose that the arbitrary sequence $\mathbf{x}_n \in \mathcal{D}$ for $1 \leq n \leq N$ is the target sequence $\mathbf{x}^F$. Then, θ_{rtr} is updated on $\mathcal{D}$ except for $\mathbf{x}_n$ from the randomly initialized parameters θ_0:

$$\theta_{\text{ptr}} = \theta_0 - \sum_{i=1}^{N} \nabla_\theta \mathcal{L}(\mathbf{x}_i; \theta_{i-1}), \tag{3}$$

$$\theta_{\text{rtr}} = \theta_0 - \sum_{i=1}^{n-1} \nabla_\theta \mathcal{L}(\mathbf{x}_i; \theta_{i-1}) - \sum_{i=n+1}^{N} \nabla_\theta \mathcal{L}(\mathbf{x}_i; \theta^*_{i-1}), \tag{4}$$

where θ^*_j refers to the parameters trained on $\{\mathbf{x}_1, \cdots, \mathbf{x}_j\}$ without the target sequence $\mathbf{x}_n$ for $n \leq j$. In other words, θ_{rtr} is equal to θ^*_N, since it is trained on $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ except for $\mathbf{x}_n$. By leveraging the equations above, we can derive the equation where θ_{rtr} is represented by θ_{ptr}:

$$\theta_{\text{rtr}} = \theta_{\text{ptr}} + \nabla_\theta \mathcal{L}(\mathbf{x}_n; \theta_{n-1}) + S, \tag{5}$$

$$S = \sum_{i=n+1}^{N} \nabla_\theta \mathcal{L}(\mathbf{x}_i; \theta_{i-1}) - \nabla_\theta \mathcal{L}(\mathbf{x}_i; \theta^*_{i-1}). \tag{6}$$

Derivation of a Tractable Solution Although the derived equation above is reasonable, we cannot compute the $\sum$ in Equation 6 because θs during training are intractable. To address this, we constrain $\mathbf{N} \approx \mathbf{n} + \mathbf{1}$, where we suppose the target sequence $\mathbf{x}_n$ is trained just before the last sequence:

$$S = \nabla_\theta \mathcal{L}(\mathbf{x}_{n+1}; \theta_n) - \nabla_\theta \mathcal{L}(\mathbf{x}_{n+1}; \theta_{n-1}), \tag{7}$$

where $\mathbf{x}_{n+1}$ refers to remaining data $\mathbf{x}^R \in \mathcal{D}^R$ without the target sequence $\mathbf{x}_n (= \mathbf{x}^F)$, and we can say that θ_n has more knowledge of the target sequence than θ_{n-1} does.

Iterative Update Equation Using Equations 5 and 7, we initialize θ_{n-1} with θ_{ptr}, which is iteratively updated to unlearn the target sequence $\mathbf{x}^F$. To assure the relationship between θ_n and θ_{n-1}, we fix θ_n as θ_{ptr}, where the parameters remain frozen during unlearning. Then, the iterative update equation for unlearning the target sequence is

$$\theta := \theta + \nabla_\theta \mathcal{L}(\mathbf{x}^F; \theta) + S, \tag{8}$$

$$S = \nabla_\theta \mathcal{L}(\mathbf{x}^R; \theta_{\text{ptr}}) - \nabla_\theta \mathcal{L}(\mathbf{x}^R; \theta), \tag{9}$$

where θ is trainable parameters initialized with θ_{ptr}, and is unlearned until convergence to θ_{rtr}.

From Gradients to Loss Terms For training, we use the following losses corresponding to the derived gradient terms:

$$\mathcal{L}_{\text{asc}} = \mathbb{E}_{\mathcal{D}^F}[\log(p_\theta(\mathbf{x}))], \tag{10}$$

$$\mathcal{L}_{\text{ret}} = \mathbb{E}_{\mathcal{D}^R}[\log(p_{\theta_{\text{ptr}}}(\mathbf{x})) - \log(p_\theta(\mathbf{x}))], \tag{11}$$

where $\mathcal{L}_{\text{asc}}$ refers to the loss for unlearning the target sequence $\mathbf{x}^F \in \mathcal{D}^F$, while $\mathcal{L}_{ret}$ denotes the loss associated with retaining the remaining data $\mathbf{x}^R \in \mathcal{D}^R$ performance. Putting everything together, the overall training objective for sequence unlearning is minimizing the following loss:

$$\mathcal{L}_{\text{pop}} = \mathcal{L}_{\text{asc}} + \lambda \mathcal{L}_{\text{ret}}, \tag{12}$$

where λ is a loss scaling hyperparameter. In $\mathcal{L}_{ret}$, the first term is ignored by the optimization, even though it contains the initial state of the pretrained LM. Since this leads to underutilization of the pretrained LM for retaining the remaining data, we use the probability distribution over the vocabulary of the pretrained LM as the soft labels. This is quite intuitive, as the objective of POP is to unlearn the target token sequence *without* deviating too much from the initial state of the pretrained LM.

3.3 Remnant Memorization Accuracy

Given a sequence of tokens $\mathbf{x} = (x_1, \ldots, x_T)$, previous studies have proposed metrics to assess "how well a model remembers a specific sequence of tokens", and unlearning can be achieved by decreasing the value of these metrics for the forgetting data. Tirumala et al. (2022) and Jang et al. (2023) suggested Memorization Accuracy (MA) and Extraction Likelihood (EL), respectively:

$$\text{MA} = \frac{\sum_{t=1}^{T-1} \mathbb{1}\{\text{argmax}(p_\theta(\cdot|x_{<t})) = x_t\}}{T-1} \tag{13}$$

$$\text{EL}_n = \frac{\sum_{t=1}^{T-n} \text{OVERLAP}_n(f_\theta(x_{<t}), x_{\geq t})}{T-n} \tag{14}$$

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

where $ng(\cdot)$ in EL represents the list of n-grams in the given sequence, and $f_\theta(x_{<t})$ represents the output sequence from the LM. As unlearning metrics are often utilized to determine the thresholds for unlearning, thereby setting the stopping point of the unlearning process, it is important that they accurately portray the privacy risk of LM post-unlearning. MA and EL, however, disregard the probabilities of tokens within the sequence. In

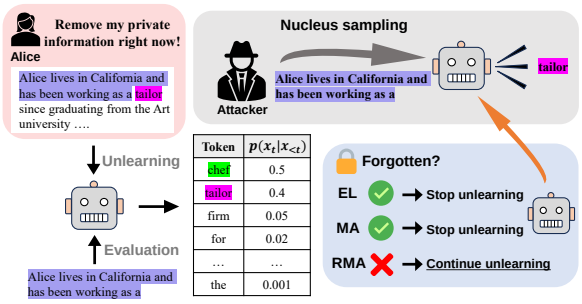


Figure 2: **Privacy Protection of RMA.** Compared to other metrics, RMA considers the token probabilities to better represent the inherent privacy risk, and provides the strongest privacy protection.

Model	Size	EL_{10}	MA	RMA
OPT	125M	4.3	40.1	31.0
	1.3B	5.9	46.4	38.4
	2.7B	6.3	47.7	39.9
GPT-Neo	125M	6.3	48.7	41.5
	1.3B	7.9	54.2	48.1
	2.7B	8.5	55.5	49.6

Table 1: Forgetting Thresholds

other words, they do not consider the situation where the target token has the second highest probability in the probability distribution for the next token prediction. When these metrics are used to determine the stopping point of the unlearning process, the resulting LM can be vulnerable to various attacks that could extract the target token through sampling methods.

To alleviate this limitation, we propose Remnant Memorization Accuracy (RMA):

$$\text{RMA} = \frac{\sum_{t=1}^{T-1} p_\theta(x_t|x_{<t})}{T-1}. \quad (15)$$

Unlike other unlearning metrics, RMA considers the probabilities of tokens to better represent the privacy risk. Models unlearned until they satisfy the forgetting thresholds for RMA are significantly less likely to be vulnerable to extraction attacks. When utilized individually, RMA is a more stringent unlearning metric, as it is more difficult to satisfy the forgetting threshold. Figure 2 shows an example of how RMA can provide a stronger privacy protection compared to other unlearning metrics. The process for obtaining the forgetting thresholds is in Section 4.4, and metric comparisons can be found in Section 5.3.

4 Experimental Setup

4.1 Baselines

We experiment on two LMs for model sizes 125M, 1.3B, 2.7B: GPT-Neo LMs (Black et al., 2022) initially pretrained on the Pile (Gao et al., 2020) corpus, and OPT LMs (Zhang et al., 2022), which are pretrained on a deduplicated version of the Pile, along with other corpora. We perform experiments with the following unlearning methods:

- **UL** (Jang et al., 2023) decreases the log-likelihood of the target token sequences – namely, only using $\mathcal{L}_{\text{asc}}$ in Equation 12.
- **POP**^{p} (Liu et al., 2022) utilizes $\mathcal{L}_{\text{asc}}$ and $\mathcal{L}_{\text{ret}}$ with the hard labels in Equation 12.
- **POP**, our main proposed method, utilizes $\mathcal{L}_{\text{asc}}$ and $\mathcal{L}_{\text{ret}}$ similarly to **POP**^{p}, where $\mathcal{L}_{\text{ret}}$ uses the probability distribution over the vocabulary of the pretrained LM as the soft labels.

In Equation 12, we set the λ as 1 for simplicity.

4.2 Target Data Curation

We source the target sequence data from the Training Data Extraction Challenge[1]. This data consists of 15,000 examples, each not exceeding 200 tokens in length. In our experiments, we construct 19 target sequence datasets, each with 32 sequences. Due to copyright issues, we randomly sample the remaining data from the uncopyrighted Pile corpus[2], without the target sequence.

4.3 Evaluation Tasks

Although POP is focused on unlearning a specific sequence of tokens, it is vital that the model performs well in all settings. Therefore, to ensure that the model is still capable of its original language modeling abilities post-unlearning, we evaluate the model on commonsense reasoning (Winogrande (Sakaguchi et al., 2021) and COPA (Gordon et al., 2012)), linguistic reasoning (Hellaswag (Zellers et al., 2019) and Lambada (Paperno et al., 2016)), and scientific reasoning (ARC-Easy (Clark et al., 2018), ARC-Challenge (Clark et al., 2018), Piqa (Bisk et al., 2020), MathQA (Amini et al., 2019) PubmedQA (Jin et al., 2019)) tasks. We also evaluate the model on dialogue tasks (Blended

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

[2] https://huggingface.co/datasets/monology/pile-uncopyrighted

Model	Method	EL_{10}	MA	RMA	Classification (Acc)	Dialogue (F1)	Epochs
OPT-125M	Pretrained	6.2	53.0	40.5	42.6	10.8	-
	UL	2.7	29.8	28.7	32.9 ±0.37	1.9 ±0.47	8.4
	POP$^\flat$	3.5	29.8	22.8	37.0 ±1.18	4.1 ±1.39	8.4
	POP	2.3	31.3	30.2	**43.3** ±0.30	**9.2** ±0.65	16.4
OPT-1.3B	Pretrained	23.1	68.4	60.6	51.5	13.3	-
	UL	2.7	32.0	30.9	36.2 ±1.74	1.8 ±1.47	5.6
	POP$^\flat$	2.1	38.4	34.3	42.4 ±0.62	5.5 ±0.57	6.2
	POP	2.3	35.6	34.4	**50.4** ±0.34	**12.3** ±0.44	7.8
OPT-2.7B	Pretrained	25.3	70.2	63.1	53.8	13.7	-
	UL	2.7	34.1	33.4	37.0 ±2.36	1.2 ±1.65	6.2
	POP$^\flat$	3.2	41.7	37.6	42.1 ±2.24	7.0 ±0.42	8.8
	POP	3.7	37.5	36.8	**52.2** ±0.35	**13.3** ±0.22	10.6
Neo-125M	Pretrained	36.1	77.9	71.1	43.5	10.0	-
	UL	2.3	45.7	39.5	40.8 ±1.87	8.0 ±1.55	10.4
	POP$^\flat$	2.2	46.2	39.4	42.9 ±0.13	10.0 ±0.29	14.6
	POP	2.6	45.8	40.4	**43.0** ±0.32	**10.4** ±0.16	13.2
Neo-1.3B	Pretrained	66.0	92.1	88.3	49.7	12.3	-
	UL	2.9	47.3	42.5	49.2 ±1.54	11.5 ±0.78	5.4
	POP$^\flat$	2.8	48.3	43.9	48.3 ±0.31	**12.1** ±0.16	6.8
	POP	3.2	48.8	44.4	**49.5** ±0.34	**12.1** ±0.19	6.0
Neo-2.7B	Pretrained	69.7	93.4	90.7	52.2	12.3	-
	UL	2.0	44.8	41.8	51.9 ±1.12	**12.3** ±0.42	6.2
	POP$^\flat$	2.8	46.6	43.3	51.8 ±0.66	12.2 ±0.17	6.4
	POP	2.2	45.9	43.0	**52.3** ±0.39	**12.3** ±0.47	6.2

Table 2: **LM Performance Comparison.** The experimental results show the average accuracy over 9 classification tasks and the average F1 over 4 dialogue tasks. POP$^\flat$ is a method that utilizes $\mathcal{L}_{asc}$ and $\mathcal{L}_{ret}$ with hard labels, and POP employs $\mathcal{L}_{asc}$ and $\mathcal{L}_{ret}$ with soft labels. The best results are **bolded**.

Skill Talk (Smith et al., 2020), Empathetic Dialogues (Rashkin et al., 2019), Wizard of Internet (Komeili et al., 2022), and Wizard of Wikipedia (Dinan et al., 2018)) to assess the generation capabilities of the model.

4.4 Forgetting Thresholds

We utilize EL_{10}, MA, and RMA to determine when to stop the unlearning process. More specifically, we consider a token sequence $\mathbf{x}^F$ to be forgotten when all three unlearning metrics fall below the average value on token sequences of Pile's evaluation set that were not seen during the pretraining. This setting was also utilized in Jang et al. (2023), where they utilized thresholds for EL_{10} and MA.[3] Table 1 shows the threshold values for each metric, and the detailed process for calculating the thresholds can be found in Appendix C.

[3]The threshold values for GPT-Neo may differ from Jang et al. (2023), as we chose to utilize the uncopyrighted version of the Pile corpus to practice ethical research. For more details, please refer to Appendix B.

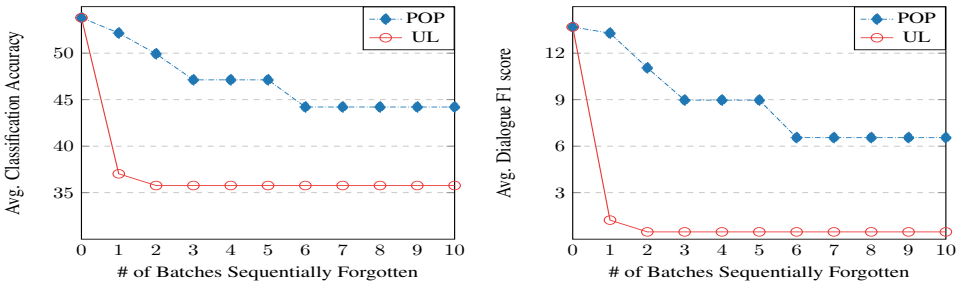
5 Results and Analyses

5.1 Main Results

We perform unlearning with 5 different random datasets of 32 target sequences, and report the averaged results for various OPT and GPT-Neo models in Table 2. Individual results can be found in Appendix E. Unlearning is performed until the model reaches the forgetting thresholds of all three metrics. The thresholds can be found in Table 1. Here are our observations:

(1) Deduplicating the pretraining corpora can reduce the privacy risks, as OPT LMs show much smaller EL_{10}, MA, and RMA values compared to the corresponding GPT-Neo models. However, deduplicating the corpora alone is not a valid unlearning solution, as the inherent privacy risk represented by EL_{10}, MA, and RMA values are not significantly lower than that of GPT-Neo.

(2) UL reaches the threshold much faster than the other two methods, demonstrated by the lower number of epochs required to reach the forgetting threshold. This is quite intuitive, as it only utilizes a single gradient ascent term, while the other two methods employ additional loss terms.



(a) Average accuracy for 9 Classification tasks.

(b) Average F1 score for 4 Dialogue tasks.

Figure 3: **Sequential Unlearning Results.** We simulate a more likely scenario of complying to numerous unlearning requests with sequential unlearning experiments. The experiments were performed on the OPT 2.7B model, and the x-axis denotes the number of batches sequentially unlearned, with each batch containing 32 target sequences. The full results for all LMs tested are available in Appendix D.

(3) The actual EL_{10}, MA, and RMA values for each model do not follow any pattern; that is, lower values do not necessarily indicate better performance. Instead, they serve as a stopping threshold to confirm the completion of unlearning target tokens.
(4) UL performs the worst in both LMs for 9 classification and 4 dialogue benchmarks, showing degradation from the initial performance. This is even more evident in the OPT models, where the drop in performance is significant for dialogue tasks, potentially showing catastrophic forgetting. POP demonstrates the least amount of degradation, representing a remarkable retention of general language modeling capabilities.
(5) UL demonstrates the largest variance in almost all benchmarks, which undermines its reliability and accentuates its dependence on the target token sequence to be unlearned.
(6) We believe that the deduplication of Pile corpus on OPT models, along with the inclusion of other corpus in the training data, contributed to the extreme degradation in UL for OPT models. As GPT-Neo is trained solely on the Pile corpus, the duplicate instances might have contributed to the retention of LM performance after unlearning with UL. As most LMs include a wide range of corpora in their training sets, we believe that this further proves the strength of POP in demonstrating optimal unlearning *and* retention of LM performance.
(7) Although POP$^\flat$outperforms UL in most benchmarks, it fails to match the performance of POP. This highlights the essential role of introducing the probability distribution over the vocabulary of the pretrained LM within $\mathcal{L}_{ret}$.

5.2 Sequential Unlearning

There are two ways to apply unlearning: batch unlearning and sequential unlearning. The results shown in Table 2 demonstrate batch unlearning results, in which all target sequences are unlearned at once. In sequential unlearning, target sequences are split into smaller batches, which are unlearned in succession. Although batch unlearning is important to consider, sequential unlearning is a more likely real-world scenario, as unlearning requests will follow a sporadic pattern, requiring a more flexible solution.

To assess the practicality of POP, we sequentially unlearn 320 target sequences, split into 10 batches. Results for other models are available in Appendix D. As shown in Fig. 3, POP demonstrates better retention of performance in both classification and dialogue tasks compared to UL. After unlearning all 320 target sequences in 10 batches with UL, the performance of the OPT 2.7B model dropped over 18% in average classification accuracy, and 13% in average dialogue F1 score. The performance degradation in the dialogue task is extreme, as the average F1 score dropped to 0.47%, demonstrating catastrophic forgetting of general LM capabilities. Furthermore, the performance in both sets of benchmarks reaches the minimum value after 2 batches, demonstrating the major flaw in UL. POP, however, only demonstrates a moderate drop, demonstrating a decrease of 9.6% for the average classification accuracy and 7.14% for the average dialogue F1 score. Fig. 1 illustrates a qualitative example of the degradation in LM from UL. After the sequential unlearning of 10 batches with UL and POP, sequences are generated for a given prefix. The generated sequence from the LM unlearned with the UL method demonstrates catastrophic degradation, while the LM unlearned with POP generates an acceptable response. UL is not a viable option, as repeated unlearning in succession with UL results in a catastrophic failure of LMs. On the other hand, POP successfully induces the LM to unlearn the target sequences *and* does not significantly im-

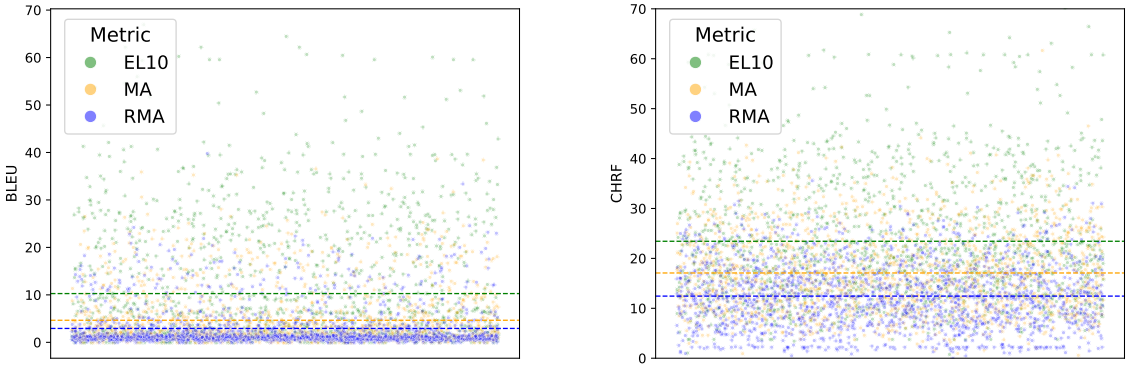


Figure 4: **Metric Comparison.** Models are unlearned until they reach the forgetting thresholds for each metric. After unlearning, we generate sequences with the resulting models, and compute BLEU and CHRF scores, where a lower score is favorable, as it indicates less overlap between the sequences. The dotted line represents the average scores for each metric. The data is spread out along the horizontal axis for visualization purposes.

Prefix	True Suffix	Metric	Generated Suffix
/* * DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER. * * Copyright ... * and Distribution License("CDDL") (collectively, the "License"). You * may not	use this file except in compliance with the License. You can * obtain a copy of the License at* http://glassfish.java.net/public/CDDL+GPL ...	EL_{10}	use this file except in compliance with the License. You can * obtain a copy of ...
		MA	use this file except in compliance with the License. You can up * to four alternative ...
		RMA	use this report file or include its work in your constitute or add any of your ...

Figure 5: Generated and True Suffixes for the given prefix. GPT-Neo LMs are unlearned with POP until the forgetting thresholds for each metric. Red indicates no unlearning, and Green indicates successful unlearning.

pact the LM performance.

5.3 Metric Analysis

We compare EL_{10}, MA, and RMA by unlearning 3 separate GPT-Neo 2.7B models with POP, and stopping the unlearning process once they reach the forgetting thresholds for each metric. We generate 50 sequences for 1 target sequence using p-sampling with probabilities of p=0.9, 0.7, and 0.5, and use the first half of the sequence as a prefix to generate the second half as a suffix. Lastly, we compare the generated and the original sequences with BLEU (Papineni et al., 2002) and CHRF (Popović, 2015), where a lower score is favorable in the context of unlearning, as it indicates less overlap between the sequences. As shown in Fig. 4, models unlearned until the RMA threshold demonstrate the lowest BLEU and CHRF scores. This proves that in the context of unlearning, RMA provides the most privacy protection, as models that satisfy the RMA threshold are less likely to generate the original sequence. We also perform a qualitative analysis, which is shown on Fig. 5. It is clear that the model unlearned until the RMA threshold demonstrates the least amount of overlap between the sequences. Models unlearned until the EL_{10} and MA thresholds, however, demonstrate some overlap in sequences, providing only partial unlearning. RMA provides the optimal privacy protection, demonstrating apt threshold for unlearning.

6 Conclusion

In this paper, we propose POP, which effectively induces the LM to unlearn target token sequences without compromising its capabilities. We demonstrate the superior performance of POP in retaining LM performance on classification and dialogue benchmarks on two different LMs for three different sizes. We also analyze a more likely scenario of complying to numerous unlearning request in succession with a sequential unlearning task, in which POP shows a much better retention of LM performance than previous work. Furthermore, we introduce RMA, a more stringent unlearning metric, and show how it can (1) better demonstrate the privacy risk of a LM, and (2) provide a stronger privacy protection when utilized to define an forgetting threshold. We hope that researchers utilize

the necessary privacy protection with POP to make LMs more viable for a wider range of tasks.

Limitations

Despite the promising performance of POP, there are areas to expand upon our work. Due to our experiments utilizing the Google Extraction benchmark, which is built on the Pile corpus, we inevitably experimented on GPT-Neo and OPT. We leave applying POP to larger models as future work. Due to the copyright issues, the forgetting threshold was determined based on the data samples chosen from the uncopyrighted Pile corpus, rather than original Pile corpus. It may result in a slight variance from previously reported the values. Furthermore, as we mentioned in Section 4.2, we sampled the remaining data from the uncopyrighted Pile corpus, which does not include high-quality data, such as the book corpus. This issue may have led to an inability to achieve further performance improvements. Lastly, we were only able to simulate the real-world setting of sequential unlearning, which at times showed no changes to the results. This may have been due to the characteristics of the Training Data Extraction Challenge, which has overlap of data sources, such as code, which follow a very distinct style. We leave the comprehensive analysis of sequential unlearning as future work to further investigate the application of sequence unlearning in LLMs.

Ethics Statement

To promote transparency within the natural language community, many have promoted the move towards removing copyrighted content from LMs. Furthermore, as the goal of our research is to improve the LLM's privacy guarantees, we were encouraged to only utilize the uncopyrighted version of the Pile corpus. All experiments were conducted on English datasets, where we looked to induce unlearning of English sequences from publicly available LMs. Utilizing the method on non-English models is not verified. Lastly, resulting models post-unlearning may generate hallucinations, which is an unintended side effect of LMs, but also an inherent problem with LMs.

Acknowledgements

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.RS-2019-II190075 Artificial Intelligence Graduate School Program(KAIST)) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2022R1A2B5B02001913) and Samsung Electronics Co., Ltd.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 308–318, New York, NY, USA. Association for Computing Machinery.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. 2022. Large-scale differentially private BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6481–6491, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tuomas Aura, Thomas A. Kuhn, and Michael Roe. 2006. Scanning electronic documents for personally identifiable information. In *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, WPES '06, page 41–50, New York, NY, USA. Association for Computing Machinery.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. 2023. Leace: Perfect linear concept erasure in closed form. *arXiv preprint arXiv:2306.03819*.

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

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.

Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. 2022. 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.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, Santa Clara, CA. USENIX Association.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.

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

Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. 2016. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg. Springer Berlin Heidelberg.

William Falcon and The PyTorch Lightning team. 2019. PyTorch Lightning.

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

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. 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, Montréal, Canada. Association for Computational Linguistics.

Chris Jay Hoofnagle, Bart Van Der Sloot, and Frederik Zuiderveen Borgesius. 2019. The european union general data protection regulation: what it is and what it means. *Information & Communications Technology Law*, 28(1):65–98.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. 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, Toronto, Canada. Association for Computational Linguistics.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 10697–10707. PMLR.

Aly Kassem, Omar Mahmoud, and Sherif Saad. 2023. Preserving privacy through dememorization: An unlearning technique for mitigating memorization risks in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4360–4379, Singapore. Association for Computational Linguistics.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8460–8478, Dublin, Ireland. Association for Computational Linguistics.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2021. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*.

Pierre Lison, Ildikó Pilán, David Sanchez, Montserrat Batet, and Lilja Øvrelid. 2021. Anonymisation

models for text data: State of the art, challenges and future directions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4188–4203, Online. Association for Computational Linguistics.

Bo Liu, Qiang Liu, and Peter Stone. 2022. Continual learning and private unlearning. In *Proceedings of The 1st Conference on Lifelong Learning Agents*, volume 199 of *Proceedings of Machine Learning Research*, pages 243–254. PMLR.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Stuart L Pardau. 2018. The california consumer privacy act: Towards a european-style privacy regime in the united states. *J. Tech. L. & Pol'y*, 23:68.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D Cotterell. 2022a. Linear adversarial concept erasure. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18400–18421. PMLR.

Shauli Ravfogel, Francisco Vargas, Yoav Goldberg, and Ryan Cotterell. 2022b. Adversarial concept erasure in kernel space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6034–6055, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2021–2030, Online. Association for Computational Linguistics.

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

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 38274–38290. Curran Associates, Inc.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System*

Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. 2021. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

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

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

Additional Details for POP

A Training Details

We conduct the experiments with the learning rate at 5e-5 with constant scheduling, and both dropout and weight decay were set to 0. We set $\lambda = 1$, the loss hyperparameter described in equation 12. We implement with Pytorch (Paszke et al., 2019) and Pytorch Lightning (Falcon and The PyTorch Lightning team, 2019). We load GPT-Neo and OPT models (125M, 1.3B, 2.7B) from Hugging Face's Transformers (Wolf et al., 2020). We utilize DeepSpeed ZeRO Stage 2 Offload and FusedAdam (Rasley et al., 2020), along with fp16 mixed precision (Micikevicius et al., 2018). The batch size is 8, and gradient accumulation is used to update all mini-batches simultaneously. During each unlearning step, we use 32 retain data for training. We use NVIDIA RTX A6000 and 3090 GPUs; the unlearning process takes approximately 1 hour for the 125M model and around 3 hours for the 1.3B and 2.7B models.

	Size	EL_{10}	MA	RMA
Ours	125M	6.3	48.7	41.5
	1.3B	7.9	54.2	48.1
	2.7B	8.5	55.5	49.6
Jang et al.	125M	5.0	29.9	-
	1.3B	5.7	33.3	-
	2.7B	5.5	34.0	-

Table 3: Threshold comparison for GPT-Neo

B Uncopyrighted Pile Corpus

The original Pile corpus (Gao et al., 2020) is not available anymore due to copyright issues. To practice ethical research, we utilized the uncopyrighted Pile corpus[4] and computed all thresholds in Appendix C. The uncopyrighted version of the Pile corpus removes Books3, BookCorpus2, OpenSubtitles, YTSubtitles, and OWT2 from the original dataset, which is a significant portion of the dataset. Although we utilized the same process in computing the thresholds as Jang et al. (2023), the removal of copyrighted data impacted the threshold values. Table 3 shows the different threshold values for GPT-Neo. Although this may have led to discrepancies between the performance of the UL method presented in Jang et al. (2023) and in our experiment for GPT-Neo model, we believe that the differences are minimal, and will not impact the relative performance of the methods.

C Measuring Forgetting Thresholds

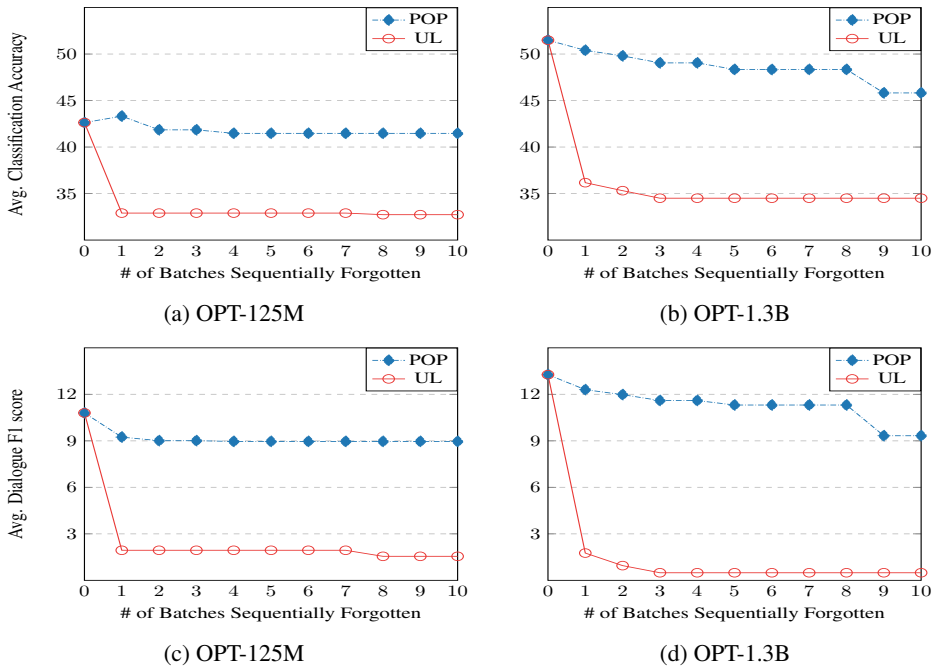
For measuring the forgetting threshold, we used the uncopyrighted Pile corpus to conduct research ethically. We sampled 10,000 data through weighted sampling based on the domain distribution of the Pile corpus. Table 4 shows the number of sampled data for each domain. We measured the thresholds for EL_{10}, MA, and RMA, and the results are presented in Table 1.

[4] https://huggingface.co/datasets/monology/pile-uncopyrighted

Domain	Number of data
Pile-CC	2739
PubMed Central	1920
ArXiv	1190
Github	1010
FreeLaw	820
StackExchange	680
USPTO Backgrounds	490
PubMed Abstracts	410
Wikipedia (en)	200
DM Mathematics	170
EuroParl	100
HackerNews	80
Gutenberg (PG-19)	60
PhilPapers	50
NIH ExPorter	40
Ubuntu IRC	21
Enron Emails	20

Table 4: The number of data used for measuring the forgetting thresholds for each domain.

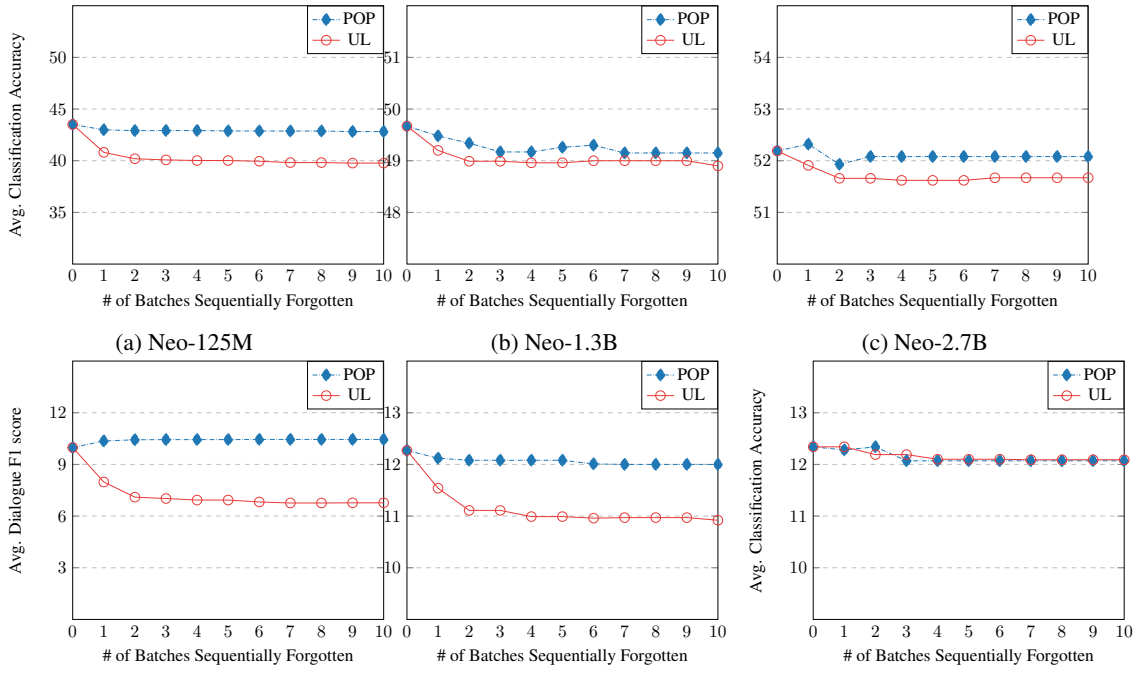
D Sequential Unlearning Results



(a) OPT-125M (b) OPT-1.3B

(c) OPT-125M (d) OPT-1.3B

Figure 6: The first row indicates the average accuracy for 9 classification tasks, and the second row shows the average F1 score for 4 Dialogue tasks for OPT models.



(a) Neo-125M (b) Neo-1.3B (c) Neo-2.7B

(d) Neo-125M (e) Neo-1.3B (f) Neo-125M

Figure 7: The first row indicates the average accuracy for 9 classification tasks, and the second row shows the average F1 score for 4 Dialogue tasks for GPT-Neo models.

E Individual Runs

	Method	Metric	Epoch	EL_{10}	MA	RMA	Lamba.	Piqa	Hella.	ARC-E	ARC-C	Copa	Wino.	MathQ	PubQ	Wiki	Inter.	Empa.	Blend.
Forgetting set0	Pretrained	-	-	9.1	58.1	45.3	38.9	62.0	28.5	45.2	20.7	66.0	53.2	21.8	47.4	11.1	12.5	9.2	10.4
	UL	EL_{10}	9	4.2	41.4	37.6	32.0	60.1	27.7	35.6	19.0	65.0	51.3	21.4	36.4	10.2	11.8	9.7	10.4
		MA	11	4.1	37.5	35.0	19.6	58.4	27.2	30.2	18.6	54.0	51.1	21.3	33.0	7.6	9.7	6.7	8.0
		RMA	14	2.8	30.3	29.0	2.6	57.0	27.1	28.8	20.0	55.0	50.7	20.6	32.4	2.2	3.0	1.2	1.7
	POP	EL_{10}	11	4.2	36.0	34.3	37.1	62.1	28.3	46.0	20.0	65.0	52.5	22.0	47.2	11.1	11.6	9.1	10.1
		MA	14	5.0	39.6	37.3	39.0	62.3	28.2	45.2	20.7	67.0	52.9	21.6	50.4	11.5	12.0	9.6	10.9
		RMA	14	2.8	31.8	30.9	39.5	62.4	28.3	44.4	21.7	68.0	52.6	21.7	53.0	9.8	11.2	7.5	9.4
	$POP^\flat$	EL_{10}	13	4.1	45.8	33.3	19.6	60.3	27.8	43.4	17.6	68.0	53.3	21.6	56.2	9.3	10.6	10.1	10.6
		MA	15	4.1	36.5	27.7	7.1	59.1	27.2	41.6	18.0	65.0	51.0	21.8	52.2	3.2	4.2	7.8	5.5
		RMA	16	4.1	36.5	27.7	7.1	59.1	27.2	41.6	18.0	65.0	51.0	21.8	52.2	3.2	4.2	7.8	5.5
Forgetting set1	Pretrained	-	-	8.2	54.4	41.5	38.9	62.0	28.5	45.2	20.7	66.0	53.2	21.8	47.4	11.1	12.5	9.2	10.4
	UL	EL_{10}	8	4.2	35.3	34.0	4.6	57.8	27.2	29.3	19.0	62.0	49.7	21.3	32.4	4.5	5.2	2.2	3.2
		MA	10	5.1	39.8	37.2	28.7	59.5	27.4	33.3	19.3	64.0	50.7	21.2	36.0	9.4	10.9	9.0	9.3
		RMA	10	3.4	28.0	27.2	0.8	57.3	26.6	28.2	21.4	54.0	49.8	20.9	32.4	2.2	2.8	1.4	1.6
	POP	EL_{10}	10	4.3	38.9	37.2	36.4	62.1	28.3	45.0	20.3	65.0	52.2	21.7	44.4	11.3	11.7	9.0	10.2
		MA	12	4.9	39.6	37.9	37.2	61.9	28.3	45.2	21.0	65.0	52.3	21.6	47.4	11.2	11.4	9.1	10.3
		RMA	12	2.6	30.2	29.3	37.8	61.4	28.3	43.0	20.7	70.0	51.5	21.7	53.6	8.0	10.4	6.7	8.4
	$POP^\flat$	EL_{10}	11	3.1	16.1	13.8	5.6	59.1	27.4	42.5	17.3	66.0	50.4	21.7	34.6	1.2	1.2	3.9	2.2
		MA	13	6.5	37.1	25.9	8.2	59.5	27.1	41.8	18.3	64.0	51.1	21.8	50.4	2.7	3.6	7.5	5.1
		RMA	13	6.4	43.7	29.1	20.3	60.4	27.7	43.9	18.3	68.0	52.6	21.7	56.0	9.0	10.2	10.0	10.6
Forgetting set2	Pretrained	-	-	3.1	50.4	38.7	38.9	62.0	28.5	45.2	20.7	66.0	53.2	21.8	47.4	11.1	12.5	9.2	10.4
	UL	EL_{10}	6	3.1	50.4	38.8	38.9	62.0	28.5	45.2	20.7	66.0	53.2	21.8	47.4	11.1	12.5	9.2	10.4
		MA	8	2.7	39.1	35.9	34.0	60.2	28.0	36.7	20.0	67.0	51.3	21.6	36.0	10.8	12	9.8	10.9
		RMA	8	2.1	30.5	29.5	5.2	57.1	27.1	29.5	19.3	59.0	49.2	20.7	32.4	2.9	3.8	1.4	2.4
	POP	EL_{10}	9	3.1	50.4	38.8	38.9	62.0	28.5	45.2	20.7	66.0	53.2	21.8	47.4	11.1	12.5	9.2	10.4
		MA	15	2.3	37.9	35.9	38.6	62.5	28.3	43.7	19.7	66.0	53.8	21.9	44.8	11.1	12	9.4	11.1
		RMA	16	1.3	30.8	29.7	40.0	62.0	28.4	44.6	22.4	67.0	54.1	21.9	51.4	9.5	10.9	7.5	9.6
	$POP^\flat$	EL_{10}	9	3.1	50.4	38.8	38.9	62.0	28.5	45.2	20.7	66.0	53.2	21.8	47.4	11.1	12.5	9.2	10.4
		MA	15	2.6	36.1	25.8	7.2	59.1	27.2	42.3	18.3	65.0	51.1	21.9	47.6	3.7	4.3	7.8	5.5
		RMA	13	2.4	42.1	28.8	19.8	60.0	27.7	43.7	18.0	68.0	52.3	21.6	54.0	10.3	11.4	10.1	11.0
Forgetting set3	Pretrained	-	-	6.5	51.3	38.2	38.9	62.0	28.5	45.2	20.7	66.0	53.2	21.8	47.4	11.1	12.5	9.2	10.4
	UL	EL_{10}	8	4.1	39.5	36.3	30.2	60.1	27.6	33.2	19.3	66.0	51.5	21.5	36.0	9.4	11.3	9.4	9.8
		MA	12	4.1	39.5	36.3	30.2	60.1	27.6	33.2	19.3	66.0	51.5	21.5	36.0	9.4	11.3	9.4	9.8
		RMA	9	3.4	31.5	30.3	2.2	57.6	27.0	29.1	21.7	56.0	50.3	20.9	32.4	1.9	2.5	1.1	1.3
	POP	EL_{10}	9	3.9	42.9	38.8	39.8	62.5	28.3	42.9	20.0	69.0	53.0	21.9	48.8	11.3	12.1	9.3	10.4
		MA	12	3.5	39.0	36.1	39.5	62.7	28.2	43.7	20.0	67.0	53.2	21.9	46.0	11.5	12.3	9.6	10.7
		RMA	12	2.9	31.7	30.2	37.9	62.5	28.3	44.3	19.7	68.0	52.7	22.1	50.8	10.4	11.7	8.4	10.1
	$POP^\flat$	EL_{10}	12	3.1	21.8	18.3	2.5	59.3	27.3	41.6	19.0	63.0	50.8	21.3	33.6	1.4	2.1	6.2	3.0
		MA	15	5.2	36.6	25.3	7.2	59.5	27.2	41.6	18.0	66.0	51.0	21.7	47.0	3.1	3.9	7.7	5.3
		RMA	15	5.5	42.8	28.4	19.9	60.3	27.7	43.6	17.6	68.0	53.0	21.7	54.4	9.7	10.8	10.0	10.8
Forgetting set4	Pretrained	-	-	4.2	50.9	38.6	38.9	62.0	28.5	45.2	20.7	66.0	53.2	21.8	47.4	11.1	12.5	9.2	10.4
	UL	EL_{10}	7	4.2	50.9	38.6	38.9	62.0	28.5	45.2	20.7	66.0	53.2	21.8	47.4	11.1	12.5	9.2	10.4
		MA	10	3.3	39.5	37.3	27.3	59.6	27.5	30.7	18.3	60.0	50.5	21.6	35.2	10.6	12.2	9.4	10.7
		RMA	11	2.0	28.6	27.5	2.2	57.3	26.7	29.8	19.7	60.0	49.7	20.8	32.4	1.5	1.8	0.9	1.2
	POP	EL_{10}	9	4.2	50.9	38.6	38.9	62.0	28.5	45.2	20.7	66.0	53.2	21.8	47.4	11.1	12.5	9.2	10.4
		MA	12	2.9	39.4	37.6	35.7	62.6	28.2	44.1	21.0	66.0	53.4	21.7	46.0	11.3	12.1	9.3	10.3
		RMA	12	1.8	32.0	30.9	38.3	62.8	28.3	43.6	21.7	67.0	53.0	22.4	55.0	9.0	10.4	7.2	8.8
	$POP^\flat$	EL_{10}	10	4.2	50.9	38.6	38.9	62.0	28.5	45.2	20.7	66.0	53.2	21.8	47.4	11.1	12.5	9.2	10.4
		MA	16	4.7	38.7	28.2	6.4	59.0	27.2	42.3	18.0	65.0	51.6	21.5	46.8	3.2	3.6	7.4	4.7
		RMA	16	4.9	44.1	29.8	19.3	60.0	27.7	43.4	18.0	68.0	52.1	21.6	53.8	9.9	11.1	10.0	11.0

Table 5: All of the individual runs for OPT 125M. The **Metric** column indicates the checkpoint at which the given metric reaches the pre-defined threshold. In Table 2, we reported the result when all metrics are satisfied with each threshold.

	Method	Metric	Epoch	EL_{10}	MA	RMA	Lamba.	Piqa	Hella.	ARC-E	ARC-C	Copa	Wino.	MathQ	PubQ	Wiki	Inter.	Empa.	Blend.
Forgetting set0	Pretrained	-	-	29.9	70.9	64.3	58.9	71.6	39.7	55.6	24.1	76	56.7	23.2	57.8	13.0	13.7	12.6	13.8
	UL	EL_{10}	5	4.3	57.7	53.1	58.2	71.2	40.0	52.6	23.7	75.0	56.8	23.4	58.0	13.0	14.6	12.6	13.9
		MA	5	4.0	39.3	36.8	20.5	58.6	32.6	30.9	23.4	64.0	51.1	21.4	47.4	4.1	4.8	2.4	4.8
		RMA	6	4.0	39.3	36.8	20.5	58.6	32.6	30.9	23.4	64.0	51.1	21.4	47.4	4.1	4.8	2.4	4.8
	POP	EL_{10}	6	5.7	55.8	53.1	60.8	71.2	39.7	52.7	25.4	76.0	56.3	23.5	58.0	13.6	13.9	12.7	13.7
		MA	6	3.7	37.9	35.9	58.2	70.8	38.4	52.4	24.4	75.0	56.1	22.7	58.0	11.7	13.5	12.3	12.8
		RMA	7	3.7	37.9	35.9	58.2	70.8	38.4	52.4	24.4	75.0	56.1	22.7	58.0	11.7	13.5	12.3	12.8
	$POP^\flat$	EL_{10}	6	5.0	51.6	43.4	26.5	66.9	31.5	48.7	21.7	73.0	54.9	22.3	57.0	8.9	10.2	10.4	11.1
		MA	7	2.2	41.2	37.8	15.3	65.3	30.9	46.0	20.3	67.0	53.1	22.5	56.4	4.6	5.9	6.3	6.7
		RMA	7	2.2	41.2	37.8	15.3	65.3	30.9	46.0	20.3	67.0	53.1	22.5	56.4	4.6	5.9	6.3	6.7
Forgetting set1	Pretrained	-	-	29.3	71.7	64.5	58.9	71.6	39.7	55.6	24.1	76.0	56.7	23.2	57.8	13.0	13.7	12.6	13.8
	UL	EL_{10}	5	5.5	51.4	48.5	27.7	62.0	33.8	33.9	23.4	60.0	53.2	20.5	55.2	8.8	9.5	7.4	9.2
		MA	5	4.0	41.4	39.7	12.7	60.0	31.6	30.3	22.0	61.0	52.5	21.6	46.4	2.7	3.4	1.6	2.5
		RMA	5	4.0	26.9	26.5	0.4	57.7	29.3	24.9	22.0	60.0	51.4	21.0	42.0	1.3	1.0	1.6	0.9
	POP	EL_{10}	5	5.1	54.5	52.0	60.3	70.8	40.0	53.4	26.4	76.0	55.6	23.4	57.6	13.1	14.0	12.5	13.4
		MA	6	3.6	40.3	38.7	59.0	70.9	38.4	51.7	25.8	75.0	55.9	22.6	57.6	11.9	13.6	12.0	12.2
		RMA	6	2.7	30.1	29.7	57.4	70.6	37.6	51.7	26.1	74.0	56.2	22.4	57.2	11.5	13.1	11.2	11.7
	$POP^\flat$	EL_{10}	6	2.4	43.5	41.2	16.7	65.6	31.1	45.0	19.3	68.0	53.8	22.2	57.0	4.4	5.6	6.7	6.7
		MA	6	2.4	43.5	41.2	16.7	65.6	31.1	45.0	19.3	68.0	53.8	22.2	57.0	4.4	5.6	6.7	6.7
		RMA	6	4.1	41.9	37.7	29.7	64.1	31.1	45.9	21.4	68.0	53.3	22.7	54.2	4.0	4.8	5.5	5.2
Forgetting set2	Pretrained	-	-	14.4	63.0	54.1	58.9	71.6	39.7	55.6	24.1	76.0	56.7	23.2	57.8	13	13.7	12.6	13.8
	UL	EL_{10}	4	4.2	54.6	50.8	59.2	70.6	40.3	52.2	25.4	75.0	56.8	23.1	57.8	13.2	14.2	12.6	14.1
		MA	5	2.2	41.1	39.6	31.0	60.3	33.0	31.0	21.4	59.0	52.2	20.8	55.4	5.7	6.7	4.4	7.1
		RMA	5	1.5	25.9	25.6	1.1	57.0	30.2	24.5	22.4	58.0	50.2	20.9	55.4	0.2	0.3	0.1	0.2
	POP	EL_{10}	5	4.5	54.3	50.6	59.6	70.6	40.4	51.9	25.8	77.0	57.0	23.8	58.0	13.1	13.8	12.6	14.1
		MA	6	1.8	44.1	42.3	59.0	70.2	38.5	52.6	24.1	73.0	55.5	23.6	57.2	12.0	13.3	12.2	12.6
		RMA	6	1.4	37.4	36.6	55.7	70.0	37.9	51.7	23.7	74.0	55.9	23.0	56.6	12.1	12.6	11.8	11.8
	$POP^\flat$	EL_{10}	6	5.5	58.4	49.9	54.2	70.6	36.6	55.2	22.7	76.0	57.0	23.0	57.2	12.4	13.2	11.9	13.2
		MA	7	1.7	41.0	36.9	16.6	65.3	30.9	46.0	20.0	68.0	53.2	22.8	56.4	4.6	5.5	7.5	7.4
		RMA	7	1.7	41.0	36.9	16.6	65.3	30.9	46.0	20.0	68.0	53.2	22.8	56.4	4.6	5.5	7.5	7.4
Forgetting set3	Pretrained	-	-	27.0	70.3	62.2	58.9	71.6	39.7	55.6	24.1	76.0	56.7	23.2	57.8	13.0	13.7	12.6	13.8
	UL	EL_{10}	4	3.4	57.7	52.3	57.0	69.8	39.4	50.3	25.4	75.0	56.3	23.1	58.0	12.7	13.9	12.2	13.6
		MA	5	2.7	37.2	35.4	12.1	59.7	31.8	30.3	21.4	57.0	50.3	21.3	47.0	2.5	2.8	1.4	2.5
		RMA	5	2.7	37.2	35.4	12.1	59.7	31.8	30.3	21.4	57.0	50.3	21.3	47.0	2.5	2.8	1.4	2.5
	POP	EL_{10}	4	5.4	59.3	54.2	57.7	70.7	40.2	52.2	24.8	77.0	57.0	23.5	57.6	12.7	14.4	12.3	13.8
		MA	5	2.8	44.8	42.5	60.3	70.8	39.4	52.0	24.4	76.0	55.9	23.2	57.8	12.9	14.0	12.6	13.5
		RMA	5	1.8	35.5	33.7	58.2	70.4	38.6	51.7	24.4	75.0	56.3	23.3	58.0	12.3	13.8	12.6	13.1
	$POP^\flat$	EL_{10}	5	4.1	48.8	40.7	31.3	67.5	32.2	49.4	20.0	73.0	54.9	22.7	56.8	9.2	11.0	10.2	11.3
		MA	6	2.3	36.0	29.0	15.7	65.3	30.6	45.5	20.3	67.0	54.9	22.6	56.2	3.8	5.0	7.0	6.5
		RMA	6	2.3	36.0	29.0	15.7	65.3	30.6	45.5	20.3	67.0	54.9	22.6	56.2	3.8	5.0	7.0	6.5
Forgetting set4	Pretrained	-	-	15.0	66.0	57.6	58.9	71.6	39.7	55.6	24.1	76.0	56.7	23.2	57.8	13.0	13.7	12.6	13.8
	UL	EL_{10}	4	2.2	54.9	51.2	58.5	70.6	40.3	51.0	24.1	77.0	56.3	23.5	57.8	12.9	14.2	12.4	14.1
		MA	6	2.3	43.9	41.8	31.1	60.8	32.6	31.4	22.0	56.0	52.2	21.3	54.6	6.4	7.6	5.3	7.5
		RMA	6	1.5	30.5	30.1	3.5	57.6	30.7	29.6	22.0	58.0	50.2	20.8	46.4	0.9	1.4	0.6	1.4
	POP	EL_{10}	5	4.5	56.1	52.3	58.8	70.6	40.3	50.6	25.1	77.0	56.6	23.6	57.8	12.7	14.3	12.4	14.1
		MA	6	2.5	44.4	42.7	59.1	70.5	38.5	51.5	25.8	77.0	56.4	23.2	57.8	12.1	13.5	11.7	12.4
		RMA	6	2.0	36.9	36.1	57.1	70.7	37.8	50.8	25.8	76.0	56.9	22.5	57.0	12.1	12.9	11.4	11.9
	$POP^\flat$	EL_{10}	6	5.0	59.9	52.2	53.9	70.6	36.6	55.4	23.1	76.0	56.8	22.8	58.0	12.4	12.9	11.6	13.0
		MA	7	2.1	42.8	41.0	12.9	65.2	30.6	46.0	20.3	68.0	52.6	22.5	56.6	3.5	4.9	6.8	6.1
		RMA	8	0.1	31.9	30.2	21.3	65.3	31.0	45.7	21.7	67.0	53.8	22.6	56.2	3.8	4.2	6.3	5.7

Table 6: All of the individual runs for OPT 1.3B. The **Metric** column indicates the checkpoint at which the given metric reaches the pre-defined threshold. In Table 2, we reported the result when all metrics are satisfied with each threshold.

	Method	Metric	Epoch	EL_{10}	MA	RMA	Lamba.	Piqa	Hella.	ARC-E	ARC-C	Copa	Wino.	MathQ	PubQ	Wiki	Inter.	Empa.	Blend.
Forgetting set0	Pretrained			32.2	72.6	66.3	64.4	74.3	43.5	56.8	27.1	78.0	59.1	23.0	58.2	13.4	14.7	13.0	13.6
	UL	EL_{10}	6	5.6	60.8	57.3	63.6	72.6	42.4	50.4	28.1	72.0	58.9	23.0	57.0	12.9	15.1	12.3	14.1
		MA	8	3.5	40.6	39.1	20.2	62.0	34.3	35.1	23.7	59.0	51.6	21.8	53.2	3.3	3.9	3.4	4.6
		RMA	8	3.5	40.6	39.1	20.2	62.0	34.3	35.1	23.7	59.0	51.6	21.8	53.2	3.3	3.9	3.4	4.6
	POP	EL_{10}	6	6.0	53.8	51.9	64.7	73.3	42.5	56.3	29.2	74.0	59.0	22.4	57.6	13.7	14.6	13.6	13.8
		MA	8	4.3	44.0	42.6	62.5	73.0	41.9	55.4	28.1	73.0	58.6	23.0	57.6	13.2	14.4	13.2	13.6
		RMA	8	4.7	36.2	35.6	60.5	72.7	41.3	55.6	27.1	73.0	58.2	23.2	57.6	12.8	14.3	13.0	13.3
	$POP^\flat$	EL_{10}	5	4.9	46.6	43.8	15.7	65.7	32.5	46.2	23.1	68.0	55.6	22.6	40.4	6.1	8.0	6.1	8.1
		MA	6	4.9	46.6	43.8	15.7	65.7	32.5	46.2	23.1	68.0	55.6	22.6	40.4	6.1	8.0	6.1	8.1
		RMA	6	3.0	36.0	34.2	35.5	66.1	33.8	49.6	23.4	67.0	55.3	22.2	51.8	7.3	8.2	7.1	7.7
Forgetting set1	Pretrained			32.1	73.8	67.8	64.4	74.3	43.5	56.8	27.1	78.0	59.1	23.0	58.2	13.4	14.7	13.0	13.6
	UL	EL_{10}	4	6.0	55.4	53.3	31.5	63.7	36.1	36.2	25.1	55.0	50.4	21.7	43.8	8.8	9.4	7.9	9.0
		MA	7	4.4	43.1	41.9	10.6	60.9	32.8	33.9	22.0	56.0	51.5	22.1	46.6	1.6	1.5	1.1	2.4
		RMA	7	4.2	30.8	30.4	0.0	57.1	29.6	28.0	20.0	50.0	51.6	19.4	56.8	0.1	0.1	0.0	0.1
	POP	EL_{10}	5	5.1	47.9	46.0	63.8	72.8	42.0	55.2	27.1	74.0	57.9	22.4	57.6	13.6	14.5	13.4	13.2
		MA	6	6.2	39.7	38.6	61.0	71.8	41.4	54.3	25.1	76.0	57.3	22.5	57.4	13.2	14.4	13.0	13.4
		RMA	6	6.2	39.7	38.6	61.0	71.8	41.4	54.3	25.1	76.0	57.3	22.5	57.4	13.2	14.4	13.0	13.4
	$POP^\flat$	EL_{10}	5	2.4	49.3	46.5	20.5	66.5	33.1	46.9	23.4	68.0	55.0	22.3	43.2	6.7	8.4	7.2	8.1
		MA	6	7.2	44.9	39.9	26.9	65.6	32.5	46.6	22.7	67.0	55.5	22.2	46.0	5.5	7.5	6.0	7.5
		RMA	6	7.2	44.9	39.9	26.9	65.6	32.5	46.6	22.7	67.0	55.5	22.2	46.0	5.5	7.5	6.0	7.5
Forgetting set2	Pretrained			16.8	65.5	57.2	64.4	74.3	43.5	56.8	27.1	78.0	59.1	23.0	58.2	13.4	14.7	13.0	13.6
	UL	EL_{10}	4	5.5	57.7	55.0	59.6	70.2	41.4	47.4	26.4	75.0	57.1	21.2	50.2	10.7	12.9	10.8	13.1
		MA	5	2.1	42.0	40.8	27.6	60.7	34.6	33.0	24.1	57.0	51.9	22.0	56.0	2.9	3.3	3.1	4.2
		RMA	5	0.7	27.2	27.1	0.1	57.1	29.8	27.9	20.7	51.0	50.4	19.9	57.8	0.0	0.0	0.0	0.0
	POP	EL_{10}	5	5.5	57.7	55.1	57.2	70.2	41.2	46.7	26.1	73.0	56.4	21.0	47.6	10.3	12.3	10.7	12.5
		MA	6	2.0	39.5	39.2	59.2	72.7	40.9	55.7	24.4	78.0	57.7	22.4	57.6	12.4	13.8	13.0	12.9
		RMA	6	2.0	39.5	39.2	59.2	72.7	40.9	55.7	24.4	78.0	57.7	22.4	57.6	12.4	13.8	13.0	12.9
	$POP^\flat$	EL_{10}	5	3.3	51.6	44.0	24.8	68.3	32.9	49.4	23.4	70.0	55.8	22.4	44.6	9.3	10.3	10.4	11.2
		MA	6	2.8	47.4	42.6	10.2	65.8	31.9	46.6	24.4	68.0	55.3	22.6	37.4	6.3	6.9	7.2	7.3
		RMA	7	1.6	41.6	39.3	24.6	66.5	33.3	48.3	23.4	69.0	55.3	22.2	44.2	6.4	7.4	6.1	7.4
Forgetting set3	Pretrained			28.8	71.5	64.5	64.4	74.3	43.5	56.8	27.1	78.0	59.1	23.0	58.2	13.4	14.7	13.0	13.6
	UL	EL_{10}	3	5.0	61.0	57.2	58.6	71.1	40.1	47.3	25.8	75.0	57.1	21.7	48.2	10.5	13.2	10.1	13.2
		MA	6	2.8	39.2	37.9	14.4	61.3	32.9	34.7	22.4	58.0	50.7	21.5	53.0	2.0	1.8	1.4	2.6
		RMA	6	2.8	39.2	37.9	14.4	61.3	32.9	34.7	22.4	58.0	50.7	21.5	53.0	2.0	1.8	1.4	2.6
	POP	EL_{10}	6	4.8	60.9	56.7	58.4	71.0	39.9	47.4	26.4	75.0	58.1	21.9	49.0	10.0	13.0	10.2	12.8
		MA	6	3.5	44.4	42.7	65.4	72.4	42.3	56.3	27.8	76.0	57.3	22.6	57.6	13.6	14.5	13.4	13.5
		RMA	6	3.2	36.1	35.2	63.0	72.0	41.3	54.9	27.8	78.0	57.9	22.4	57.6	13.2	14.1	13.3	13.4
	$POP^\flat$	EL_{10}	4	4.6	52.6	45.7	29.2	69.5	33.4	51.5	23.4	72.0	55.6	22.3	49.0	9.1	10.3	10.0	11.3
		MA	7	1.0	43.2	40.2	9.1	65.8	31.6	46.4	24.1	66.0	55.5	22.3	39.6	7.0	7.7	8.4	8.2
		RMA	7	1.2	38.8	35.0	10.1	65.1	31.7	46.4	24.8	66.0	54.9	22.2	47.6	5.5	6.8	7.0	7.6
Forgetting set4	Pretrained			16.5	67.5	59.5	64.4	74.3	43.5	56.8	27.1	78.0	59.1	23.0	58.2	13.4	14.7	13.0	13.6
	UL	EL_{10}	4	3.6	55.6	52.6	62.4	71.4	41.5	49.9	28.5	73.0	58.8	22.1	55.6	12.4	14.9	11.5	14.0
		MA	5	3.4	45.0	43.7	30.0	63.2	35.4	34.9	26.8	55.0	50.6	21.6	50.4	4.0	4.6	4.2	5.9
		RMA	5	2.4	32.8	32.3	4.2	58.2	31.9	30.0	21.7	56.0	51.9	20.5	55.2	0.2	0.4	0.2	0.4
	POP	EL_{10}	4	3.6	55.6	52.6	62.4	71.4	41.5	49.9	28.5	73.0	58.8	22.1	55.6	12.4	14.9	11.5	14.0
		MA	5	2.4	42.7	41.6	61.2	73.0	41.5	54.9	27.5	75.0	58.1	22.8	57.4	12.9	13.7	13.4	13.4
		RMA	5	2.5	36.2	35.7	58.8	72.9	40.4	55.7	25.1	76.0	58.4	22.9	57.6	12.4	13.7	13.1	13.1
	$POP^\flat$	EL_{10}	4	5.3	60.6	52.3	56.2	73.6	39.7	55.7	24.8	79.0	57.4	22.5	59.2	12.6	13.2	12.3	13.0
		MA	6	2.8	47.5	39.7	6.6	65.1	30.8	45.7	22.7	67.0	54.7	22.1	36.6	6.0	7.2	8.0	8.5
		RMA	6	2.8	47.5	39.7	6.6	65.1	30.8	45.7	22.7	67.0	54.7	22.1	36.6	6.0	7.2	8.0	8.5

Table 7: All of the individual runs for OPT 2.7B. The **Metric** column indicates the checkpoint at which the given metric reaches the pre-defined threshold. In Table 2, we reported the result when all metrics are satisfied with each threshold.

	Method	Metric	Epoch	EL_{10}	MA	RMA	Lamba.	Piqa	Hella.	ARC-E	ARC-C	Copa	Wino.	MathQ	PubQ	Wiki	Inter.	Empa.	Blend.
Forgetting set0	Pretrained	-	-	41.5	80.5	74.0	37.6	63.4	28.2	45.7	22.0	63.0	51.5	22.5	57.6	10.5	11.3	8.4	9.7
	UL	EL_{10}	9	5.0	56.5	51.9	27.5	61.2	28.0	44.3	23.1	62.0	51.1	22.1	57.4	9.6	10.3	8.1	8.3
		MA	11	3.1	47.5	43.3	20.5	61.0	27.8	42.9	24.1	61.0	51.1	21.9	54.0	8.9	9.5	7.7	7.8
		RMA	14	2.9	42.7	38.5	17.8	60.7	27.8	41.8	23.7	61.0	51.2	21.9	53.4	8.4	9.3	7.4	7.5
	POP	EL_{10}	11	5.3	57.2	52.2	36.4	63.4	28.1	43.9	20.7	60.0	52.1	22.5	57.6	10.9	11.4	9.0	9.8
		MA	14	1.9	44.9	40.1	37.8	63.3	28.0	43.2	20.0	60.0	51.9	22.4	57.4	10.9	11.7	9.0	9.7
		RMA	14	1.9	44.9	40.1	37.8	63.3	28.0	43.2	20.0	60.0	51.9	22.4	57.4	10.9	11.7	9.0	9.7
	$POP^\flat$	EL_{10}	13	4.1	56.5	51.1	37.5	63.1	28.2	43.9	20.0	60.0	52.3	22.8	57.4	10.7	11.3	7.9	9.4
		MA	15	2.6	48.2	42.7	37.4	63.1	28.2	43.0	21.0	60.0	52.3	23.0	57.2	10.7	11.2	7.8	9.4
		RMA	16	1.9	43.7	38.4	37.0	62.8	28.1	42.9	20.3	60.0	52.5	23.0	57.2	10.6	11.3	7.7	9.3
Forgetting set1	Pretrained	-	-	36.3	75.1	68.4	37.6	63.4	28.2	45.7	22.0	63.0	51.5	22.5	57.6	10.5	11.3	8.4	9.7
	UL	EL_{10}	8	4.5	56.8	51.8	13.8	61.3	28.1	43.2	22.0	63.0	49.8	21.5	57.2	9.3	9.8	8.9	9.0
		MA	10	2.7	45.7	41.1	7.3	60.6	27.9	42.3	22.7	60.0	49.6	21.3	53.8	8.8	9.4	7.9	8.2
		RMA	10	2.7	45.7	41.1	7.3	60.6	27.9	42.3	22.7	60.0	49.6	21.3	53.8	8.8	9.4	7.9	8.2
	POP	EL_{10}	10	5.8	53.0	47.2	35.5	63.1	28.3	44.3	20.3	64.0	51.5	21.8	57.6	11.3	11.3	9.0	9.9
		MA	12	4.6	45.8	40.6	35.5	63.4	28.3	44.4	20.3	63.0	51.5	21.7	57.6	11.4	11.3	9.0	10.0
		RMA	12	4.6	45.8	40.6	35.5	63.4	28.3	44.4	20.3	63.0	51.5	21.7	57.6	11.4	11.3	9.0	10.0
	$POP^\flat$	EL_{10}	11	5.6	55.3	48.8	35.4	62.4	28.1	45.0	22.0	63.0	51.8	22.5	57.6	11.3	11.4	8.1	9.6
		MA	13	4.5	46.3	40.2	36.1	62.7	28.1	44.3	21.4	63.0	51.4	22.2	57.6	11.4	11.3	8.0	9.5
		RMA	13	4.5	46.3	40.2	36.1	62.7	28.1	44.3	21.4	63.0	51.4	22.2	57.6	11.4	11.3	8.0	9.5
Forgetting set2	Pretrained	-	-	31.7	77.6	70.4	37.6	63.4	28.2	45.7	22.0	63.0	51.5	22.5	57.6	10.5	11.3	8.4	9.7
	UL	EL_{10}	6	5.4	58.4	50.1	25.9	61.7	28.0	43.9	20.7	61.0	50.5	21.7	57.6	10.2	10.4	8.3	8.6
		MA	8	1.4	46.8	38.6	18.7	60.6	28.0	42.3	22.7	62.0	50.4	21.5	57.0	8.8	9.2	7.7	7.8
		RMA	8	1.4	46.8	38.6	18.7	60.6	28.0	42.3	22.7	62.0	50.4	21.5	57.0	8.8	9.2	7.7	7.8
	POP	EL_{10}	9	5.9	58.8	52.3	36.5	62.9	28.3	44.4	20.0	63.0	51.6	22.0	57.6	10.7	11.2	8.6	9.5
		MA	15	1.9	47.6	42.3	39.0	62.5	28.3	43.7	21.4	65.0	51.7	22.4	57.6	11.4	11.5	9.4	9.9
		RMA	16	1.6	45.3	39.9	39.0	62.9	28.3	43.2	21.4	65.0	52.0	22.4	57.6	11.5	11.5	9.5	9.9
	$POP^\flat$	EL_{10}	9	6.2	61.1	52.8	35.9	62.9	28.3	44.4	20.7	62.0	51.9	22.2	57.6	10.7	11.3	8.4	9.8
		MA	15	1.3	48.2	39.7	37.5	62.8	28.3	43.6	20.3	63.0	51.7	22.2	57.6	10.6	11.3	8.5	9.4
		RMA	13	1.7	50.0	41.4	37.7	62.7	28.2	43.9	20.0	63.0	51.6	22.2	57.6	10.6	11.6	8.4	9.4
Forgetting set3	Pretrained	-	-	37.8	76.0	68.6	37.6	63.4	28.2	45.7	22.0	63.0	51.5	22.5	57.6	10.5	11.3	8.4	9.7
	UL	EL_{10}	8	4.9	54.9	46.9	48.7	60.4	27.9	40.9	20.3	58.0	52.1	22.3	57.6	8.1	8.0	7.6	5.6
		MA	12	1.3	43.1	36.1	48.5	60.3	27.6	39.2	21.0	50.0	52.7	21.9	57.6	3.3	3.9	3.6	2.0
		RMA	9	2.1	49.3	41.5	48.8	60.2	27.7	39.2	21.0	54.0	52.4	22.0	57.6	5.7	6.1	5.8	3.7
	POP	EL_{10}	9	5.5	57.8	51.5	36.2	62.8	28.3	45.7	21.4	62.0	51.7	22.4	57.6	10.6	11.6	8.4	9.5
		MA	12	2.1	46.4	40.1	35.3	62.6	28.3	45.5	21.0	62.0	52.2	22.2	57.6	11.1	11.2	8.6	9.8
		RMA	12	2.1	46.4	40.1	35.3	62.6	28.3	45.5	21.0	62.0	52.2	22.2	57.6	11.1	11.2	8.6	9.8
	$POP^\flat$	EL_{10}	12	3.4	56.6	49.7	35.7	63.0	28.3	44.6	21.4	62.0	52.3	22.2	57.6	10.7	11.2	7.9	9.2
		MA	15	1.0	45.3	38.2	36.0	63.3	28.4	43.6	22.0	61.0	51.9	22.3	57.4	10.7	11.1	7.8	9.1
		RMA	15	1.0	45.3	38.2	36.0	63.3	28.4	43.6	22.0	61.0	51.9	22.3	57.4	10.7	11.1	7.8	9.1
Forgetting set4	Pretrained	-	-	33.1	80.1	74.1	37.6	63.4	28.2	45.7	22.0	63.0	51.5	22.5	57.6	10.5	11.3	8.4	9.7
	UL	EL_{10}	7	5.5	61.3	55.2	35.1	63.0	28.3	44.1	21.4	67.0	51.9	22.1	57.6	10.6	10.9	9.3	9.2
		MA	10	2.4	48.0	41.8	31.7	62.5	28.2	43.9	22.0	67.0	51.7	21.9	57.6	9.9	10.8	8.8	8.3
		RMA	11	2.2	43.9	37.7	31.2	62.5	28.1	43.4	22.0	67.0	51.9	21.8	57.6	9.9	10.8	8.6	8.3
	POP	EL_{10}	9	4.3	58.3	52.9	35.4	63.2	28.3	44.6	20.7	61.0	52.3	22.1	57.6	11.0	11.4	9.0	9.7
		MA	12	2.8	46.7	41.1	36.0	63.2	28.3	44.8	20.7	61.0	52.8	22.1	57.6	11.1	11.4	9.0	9.7
		RMA	12	2.8	46.7	41.1	36.0	63.2	28.3	44.8	20.7	61.0	52.8	22.1	57.6	11.1	11.4	9.0	9.7
	$POP^\flat$	EL_{10}	10	5.3	61.0	54.2	34.9	63.1	28.3	43.9	22.4	62.0	52.0	22.2	57.6	11.3	11.4	8.7	9.6
		MA	16	1.6	45.9	38.7	35.0	62.7	28.5	43.4	21.7	62.0	52.9	22.0	57.6	11.4	11.6	9.1	9.5
		RMA	16	1.6	45.9	38.7	35.0	62.7	28.5	43.4	21.7	62.0	52.9	22.0	57.6	11.4	11.6	9.1	9.5

Table 8: All of the individual runs for GPT-Neo 125M. The **Metric** column indicates the checkpoint at which the given metric reaches the pre-defined threshold. In Table 2, we reported the result when all metrics are satisfied with each threshold.

	Method	Metric	Epoch	EL_{10}	MA	RMA	Lamba.	Piqa	Hella.	ARC-E	ARC-C	Copa	Wino.	MathQ	PubQ	Wiki	Inter.	Empa.	Blend.
Forgetting set0	Pretrained	-	-	66.1	91.8	88.1	57.2	70.4	37.0	56.6	25.8	70.0	54.6	21.9	53.6	12.7	13.8	10.5	12.1
	UL	EL_{10}	5	3.6	53.1	48.4	65.6	70.4	37.3	56.8	26.1	68.0	56.2	21.9	55.4	12.4	12.6	9.9	10.8
		MA	5	3.6	53.1	48.4	65.6	70.4	37.3	56.8	26.1	68.0	56.2	21.9	55.4	12.4	12.6	9.9	10.8
		RMA	6	2.1	42.1	37.0	67.1	70.2	37.3	55.6	26.1	68.0	56.7	22.0	55.6	11.3	11.4	9.1	10.1
	POP	EL_{10}	6	4.1	53.7	49.4	56.8	71.0	37.3	56.1	26.1	70.0	54.8	21.7	53.6	12.6	14.0	10.5	12.2
		MA	6	4.1	53.7	49.4	56.8	71.0	37.3	56.1	26.1	70.0	54.8	21.7	53.6	12.6	14.0	10.5	12.2
		RMA	7	2.2	46.4	42.1	55.6	70.7	37.4	56.1	26.8	71.0	55.3	21.9	54.4	12.4	13.8	10.6	12.3
	$POP^\flat$	EL_{10}	6	4.3	57.9	53.7	58.0	70.7	37.6	53.6	25.8	70.0	54.1	21.3	45.4	12.7	13.2	10.3	11.7
		MA	7	2.5	47.8	43.9	56.6	70.7	37.6	53.6	25.4	70.0	53.8	21.7	44.2	12.8	13.6	10.4	11.7
		RMA	7	2.5	47.8	43.9	56.6	70.7	37.6	53.6	25.4	70.0	53.8	21.7	44.2	12.8	13.6	10.4	11.7
Forgetting set1	Pretrained	-	-	68.3	92.1	87.8	57.2	70.4	37.0	56.6	25.8	70.0	54.6	21.9	53.6	12.7	13.8	10.5	12.1
	UL	EL_{10}	5	5.7	50.9	46.6	57.9	69.9	37.3	55.7	23.7	68.0	55.3	21.3	53.4	11.5	12.7	8.9	10.7
		MA	5	5.7	50.9	46.6	57.9	69.9	37.3	55.7	23.7	68.0	55.3	21.3	53.4	11.5	12.7	8.9	10.7
		RMA	5	5.7	50.9	46.6	57.9	69.9	37.3	55.7	23.7	68.0	55.3	21.3	53.4	11.5	12.7	8.9	10.7
	POP	EL_{10}	5	7.9	57.0	53.0	56.4	69.9	37.3	56.4	25.4	71.0	54.5	21.5	53.0	12.4	13.7	9.8	11.6
		MA	6	5.8	46.3	42.7	56.8	69.6	37.5	56.4	25.4	71.0	54.3	21.4	53.4	12.4	13.7	9.6	11.6
		RMA	6	5.8	46.3	42.7	56.8	69.6	37.5	56.4	25.4	71.0	54.3	21.4	53.4	12.4	13.7	9.6	11.6
	$POP^\flat$	EL_{10}	6	5.8	51.4	47.0	57.2	70.2	37.0	55.0	26.4	71.0	55.1	21.2	42.2	12.9	13.1	9.9	11.4
		MA	6	5.8	51.4	47.0	57.2	70.2	37.0	55.0	26.4	71.0	55.1	21.2	42.2	12.9	13.1	9.9	11.4
		RMA	6	5.8	51.4	47.0	57.2	70.2	37.0	55.0	26.4	71.0	55.1	21.2	42.2	12.9	13.1	9.9	11.4
Forgetting set2	Pretrained	-	-	63.0	90.8	87.4	57.2	70.4	37.0	56.6	25.8	70.0	54.6	21.9	53.6	12.7	13.8	10.5	12.1
	UL	EL_{10}	4	6.6	61.5	57.7	44.7	70.1	37.1	55.6	25.8	74.0	53.9	21.3	45.6	12.4	13.3	11.1	11.6
		MA	5	1.7	52.1	47.9	40.9	70.0	37.1	54.9	25.1	74.0	53.8	21.5	43.8	12.3	13.1	11.1	11.4
		RMA	5	1.7	52.1	47.9	40.9	70.0	37.1	54.9	25.1	74.0	53.8	21.5	43.8	12.3	13.1	11.1	11.4
	POP	EL_{10}	5	5.4	58.2	54.2	53.2	70.5	37.4	57.3	26.4	68.0	54.4	21.5	52.0	12.4	13.7	10.5	12.2
		MA	6	1.6	49.0	44.7	55.0	70.1	37.5	57.0	27.1	68.0	54.7	21.3	49.8	12.4	14.0	10.5	11.8
		RMA	6	1.6	49.0	44.7	55.0	70.1	37.5	57.0	27.1	68.0	54.7	21.3	49.8	12.4	14.0	10.5	11.8
	$POP^\flat$	EL_{10}	6	3.8	57.0	52.6	55.8	70.2	37.4	54.5	26.4	72.0	54.5	21.0	41.8	13.1	13.6	10.4	11.9
		MA	7	1.4	48.9	44.5	54.8	70.4	37.2	54.0	26.4	73.0	55.1	20.9	39.4	12.8	13.6	10.4	11.9
		RMA	7	1.4	48.9	44.5	54.8	70.4	37.2	54.0	26.4	73.0	55.1	20.9	39.4	12.8	13.6	10.4	11.9
Forgetting set3	Pretrained	-	-	66.6	92.6	88.4	57.2	70.4	37.0	56.6	25.8	70.0	54.6	21.9	53.6	12.7	13.8	10.5	12.1
	UL	EL_{10}	4	6.7	59.8	55.9	57.8	70.3	37.1	55.7	26.1	69.0	54.5	21.7	53.4	12.8	13.8	10.6	11.8
		MA	5	2.1	46.0	42.0	58.2	70.0	37.3	54.7	24.4	67.0	54.5	21.8	54.2	12.7	13.6	10.5	11.8
		RMA	5	2.1	46.0	42.0	58.2	70.0	37.3	54.7	24.4	67.0	54.5	21.8	54.2	12.7	13.6	10.5	11.8
	POP	EL_{10}	4	7.1	60.9	56.6	56.0	70.8	37.2	57.3	25.4	69.0	54.7	21.4	53.4	12.5	13.5	10.3	12.0
		MA	5	2.8	48.9	44.4	56.6	70.5	37.3	57.3	25.1	69.0	54.6	21.4	53.0	12.3	13.6	10.4	12.0
		RMA	5	2.8	48.9	44.4	56.6	70.5	37.3	57.3	25.1	69.0	54.6	21.4	53.0	12.3	13.6	10.4	12.0
	$POP^\flat$	EL_{10}	5	6.6	60.1	55.7	58.8	70.2	37.3	54.1	26.1	70.0	54.0	21.6	46.2	12.8	13.5	10.2	12.1
		MA	6	2.2	47.5	43.2	57.8	70.1	37.3	54.5	26.8	71.0	54.1	21.2	44.0	12.8	13.4	10.2	12.0
		RMA	6	2.2	47.5	43.2	57.8	70.1	37.3	54.5	26.8	71.0	54.1	21.2	44.0	12.8	13.4	10.2	12.0
Forgetting set4	Pretrained	-	-	66.1	93.3	89.9	57.2	70.4	37.0	56.6	25.8	70.0	54.6	21.9	53.6	12.7	13.8	10.5	12.1
	UL	EL_{10}	4	6.3	62.7	57.1	61.5	70.1	36.2	54.3	25.4	72.0	55.3	22.1	55.4	12.9	13.5	11.0	12.1
		MA	6	3.0	45.4	39.0	63.7	69.2	35.5	52.7	24.4	71.0	55.0	21.8	56.8	13.2	13.2	10.6	11.4
		RMA	6	3.0	45.4	39.0	63.7	69.2	35.5	52.7	24.4	71.0	55.0	21.8	56.8	13.2	13.2	10.6	11.4
	POP	EL_{10}	5	5.4	61.3	56.1	55.4	70.2	37.1	57.0	26.4	73.0	55.0	21.5	53.8	12.6	13.5	10.8	12.2
		MA	6	3.4	53.3	47.9	54.5	69.8	37.1	57.3	26.4	71.0	54.9	21.8	53.4	12.4	13.6	10.8	12.3
		RMA	6	3.4	53.3	47.9	54.5	69.8	37.1	57.3	26.4	71.0	54.9	21.8	53.4	12.4	13.6	10.8	12.3
	$POP^\flat$	EL_{10}	6	4.1	61.5	56.5	57.9	70.1	37.0	53.8	24.8	73.0	54.5	21.5	49.0	13.1	13.6	11.0	11.6
		MA	7	3.4	54.1	48.4	56.6	70.1	37.1	53.8	25.8	73.0	53.9	21.6	46.8	13.3	13.3	11.1	11.7
		RMA	8	2.1	45.9	41.0	55.9	69.9	37.1	53.8	25.4	74.0	53.9	21.5	46.8	13.1	13.2	11.0	11.6

Table 9: All of the individual runs for GPT-Neo 1.3B. The **Metric** column indicates the checkpoint at which the given metric reaches the pre-defined threshold. In Table 2, we reported the result when all metrics are satisfied with each threshold.

	Method	Metric	Epoch	EL_{10}	MA	RMA	Lamba.	Piqa	Hella.	ARC-E	ARC-C	Copa	Wino.	MathQ	PubQ	Wiki	Inter.	Empa.	Blend.
Forgetting set0	Pretrained	-	-	68.3	92.7	90.0	62.3	73.0	40.7	59.8	25.4	74.0	56.1	21.4	57.0	12.4	13.7	10.9	12.4
	UL	EL_{10}	6	3.9	56.0	53.3	64.8	73.2	41.1	58.6	29.8	72.0	55.2	21.3	57.4	13.1	13.8	11.7	12.5
		MA	8	2.5	37.2	34.9	65.5	73.2	41.3	57.7	29.2	72.0	55.5	21.4	57.4	13.2	13.7	11.3	12.3
		RMA	8	2.5	37.2	34.9	65.5	73.2	41.3	57.7	29.2	72.0	55.5	21.4	57.4	13.2	13.7	11.3	12.3
	POP	EL_{10}	6	6.7	61.1	58.9	63.5	73.1	41.3	58.9	28.1	73.0	54.9	21.4	57.4	12.3	13.3	10.4	12.0
		MA	8	2.3	44.7	42.8	62.6	73.2	41.2	58.6	28.5	72.0	54.5	21.4	57.4	12.6	13.4	10.6	12.4
		RMA	8	2.3	44.7	42.8	62.6	73.2	41.2	58.6	28.5	72.0	54.5	21.4	57.4	12.6	13.4	10.6	12.4
	$POP^\flat$	EL_{10}	5	7.4	60.4	57.2	62.7	73.3	41.1	57.9	28.8	76.0	55.6	21.0	55.6	12.5	13.2	10.8	12.5
		MA	6	2.4	46.2	42.8	61.8	73.0	41.1	58.2	27.8	76.0	55.6	21.2	54.6	12.7	13.4	10.9	12.1
		RMA	6	2.4	46.2	42.8	61.8	73.0	41.1	58.2	27.8	76.0	55.6	21.2	54.6	12.7	13.4	10.9	12.1
Forgetting set1	Pretrained	-	-	75.6	93.8	90.9	62.3	73.0	40.7	59.8	25.4	74.0	56.1	21.4	57.0	12.4	13.7	10.9	12.4
	UL	EL_{10}	4	5.8	60.4	57.1	62.8	72.4	40.9	61.0	26.1	74.0	55.1	21.7	56.6	12.9	13.3	10.8	12.0
		MA	7	2.5	49.3	46.4	62.6	72.7	41.1	61.7	25.8	70.0	55.3	21.7	57.2	12.3	13.0	10.5	11.8
		RMA	7	2.5	49.3	46.4	62.6	72.7	41.1	61.7	25.8	70.0	55.3	21.7	57.2	12.3	13.0	10.5	11.8
	POP	EL_{10}	5	6.8	56.4	52.9	60.8	73.1	41.1	60.1	27.1	73.0	54.5	21.5	56.8	12.4	13.2	10.4	11.6
		MA	6	3.2	43.1	40.4	62.0	72.9	41.3	59.4	27.5	74.0	54.9	21.7	57.0	12.3	12.8	10.2	11.7
		RMA	6	3.2	43.1	40.4	62.0	72.9	41.3	59.4	27.5	74.0	54.9	21.7	57.0	12.3	12.8	10.2	11.7
	$POP^\flat$	EL_{10}	5	7.9	62.5	59.0	62.2	72.9	41.1	57.0	26.4	74.0	54.9	21.0	53.2	12.3	13.4	10.7	12.3
		MA	6	5.3	48.9	45.8	61.0	73.0	41.1	57.1	26.4	74.0	55.2	20.9	51.6	12.3	13.1	10.9	12.5
		RMA	6	5.3	48.9	45.8	61.0	73.0	41.1	57.1	26.4	74.0	55.2	20.9	51.6	12.3	13.1	10.9	12.5
Forgetting set2	Pretrained	-	-	69.2	93.0	90.4	62.3	73.0	40.7	59.8	25.4	74.0	56.1	21.4	57.0	12.4	13.7	10.9	12.4
	UL	EL_{10}	4	6.1	60.3	57.5	51.9	72.6	41.3	58.7	27.1	73.0	54.4	21.4	57.4	12.7	13.0	11.5	12.2
		MA	5	1.3	48.4	46.2	47.6	72.2	41.8	57.0	26.4	73.0	55.0	21.7	55.8	12.4	12.3	11.5	11.7
		RMA	5	1.3	48.4	46.2	47.6	72.2	41.8	57.0	26.4	73.0	55.0	21.7	55.8	12.4	12.3	11.5	11.7
	POP	EL_{10}	5	4.1	57.2	54.4	59.5	73.0	41.3	59.1	26.8	73.0	55.9	21.2	57.0	13.1	14.0	11.3	12.5
		MA	6	1.4	48.4	46.2	61.0	72.9	41.3	58.2	26.1	73.0	56.8	21.7	57.0	13.1	13.9	11.0	12.5
		RMA	6	1.4	48.4	46.2	61.0	72.9	41.3	58.2	26.1	73.0	56.8	21.7	57.0	13.1	13.9	11.0	12.5
	$POP^\flat$	EL_{10}	5	5.8	61.9	58.9	62.4	73.3	41.0	58.0	26.1	75.0	56.0	21.2	55.2	12.3	13.6	10.8	12.2
		MA	6	1.9	53.6	51.0	60.2	73.2	41.0	58.2	25.1	73.0	55.7	21.3	53.8	12.5	14.1	11.0	11.7
		RMA	7	0.7	43.6	41.6	58.3	72.9	40.7	57.5	25.1	74.0	55.5	21.4	54.0	12.5	13.3	10.9	11.6
Forgetting set3	Pretrained	-	-	69.1	93.3	90.2	62.3	73.0	40.7	59.8	25.4	74.0	56.1	21.4	57.0	12.4	13.7	10.9	12.4
	UL	EL_{10}	3	8.0	62.3	57.9	63.1	72.7	41.1	59.3	27.1	71.0	56.1	21.4	56.8	12.7	14.0	11.2	12.8
		MA	6	1.9	44.6	40.9	64.2	72.6	41.1	57.7	27.1	70.0	56.3	21.5	57.2	12.5	13.6	11.0	12.2
		RMA	6	1.9	44.6	40.9	64.2	72.6	41.1	57.7	27.1	70.0	56.3	21.5	57.2	12.5	13.6	11.0	12.2
	POP	EL_{10}	6	2.4	48.5	44.6	62.6	73.0	41.1	58.6	27.8	72.0	55.9	21.6	56.8	12.1	13.2	10.4	11.8
		MA	6	2.4	48.5	44.6	62.6	73.0	41.1	58.6	27.8	72.0	55.9	21.6	56.8	12.1	13.2	10.4	11.8
		RMA	6	2.4	48.5	44.6	62.6	73.0	41.1	58.6	27.8	72.0	55.9	21.6	56.8	12.1	13.2	10.4	11.8
	$POP^\flat$	EL_{10}	4	4.3	60.0	56.0	62.5	73.0	41.3	57.7	27.5	78.0	55.6	21.7	55.0	12.0	13.7	10.8	12.3
		MA	7	2.0	43.8	39.9	62.9	72.9	41.4	57.8	27.1	77.0	56.0	21.7	55.4	12.0	13.5	10.7	12.2
		RMA	7	2.0	43.8	39.9	62.9	72.9	41.4	57.8	27.1	77.0	56.0	21.7	55.4	12.0	13.5	10.7	12.2
Forgetting set4	Pretrained	-	-	66.6	94.1	91.9	62.3	73.0	40.7	59.8	25.4	74.0	56.1	21.4	57.0	12.4	13.7	10.9	12.4
	UL	EL_{10}	4	3.4	56.7	53.0	65.5	72.9	40.9	58.6	26.8	75.0	56.9	21.8	57.6	13.4	13.8	12.2	12.8
		MA	5	1.8	44.7	40.8	66.8	72.9	40.9	60.0	27.1	73.0	56.7	21.9	57.6	13.3	13.3	12.4	12.6
		RMA	5	1.8	44.7	40.8	66.8	72.9	40.9	60.0	27.1	73.0	56.7	21.9	57.6	13.3	13.3	12.4	12.6
	POP	EL_{10}	4	6.2	61.7	58.5	61.5	72.9	41.3	59.1	27.8	76.0	55.9	22.1	57.0	12.7	13.9	11.0	12.3
		MA	5	3.6	51.0	47.9	62.6	73.2	41.3	59.4	27.5	75.0	56.1	21.8	57.0	12.6	13.5	10.8	12.2
		RMA	5	1.8	44.7	40.8	66.8	72.9	40.9	60.0	27.1	73.0	56.7	21.9	57.6	13.3	13.3	12.4	12.6
	$POP^\flat$	EL_{10}	4	8.0	67.1	63.4	62.4	73.1	41.1	57.7	24.8	79.0	57.1	21.5	55	12.4	13.3	10.8	12.2
		MA	6	3.8	50.4	46.5	61.5	73.0	41.2	57.5	25.8	79.0	56.0	22.0	54.0	12.6	13.9	11.2	12.2
		RMA	6	3.8	50.4	46.5	61.5	73.0	41.2	57.5	25.8	79.0	56.0	22.0	54.0	12.6	13.9	11.2	12.2

Table 10: All of the individual runs for GPT-Neo 2.7B. The **Metric** column indicates the checkpoint at which the given metric reaches the pre-defined threshold. In Table 2, we reported the result when all metrics are satisfied with each threshold.