

Learning-Time Encoding Shapes Unlearning in LLMs

Ruihan Wu*
UC, San Diego
ruw076@ucsd.edu

Konstantin Garov*
UC, San Diego
kgarov@ucsd.edu

Kamalika Chaudhuri
UC, San Diego
kamalika@ucsd.edu

Abstract

As large language models (LLMs) are increasingly deployed in the real world, the ability to “unlearn”, or remove specific pieces of knowledge post hoc, has become essential for a variety of reasons ranging from privacy regulations to correcting outdated or harmful content. Prior work has proposed unlearning benchmarks and algorithms, and has typically assumed that the training process and the target model are fixed. In this work, we empirically investigate how learning-time choices in knowledge encoding impact the effectiveness of unlearning factual knowledge. Our experiments reveal two key findings: (1) learning with paraphrased descriptions improves unlearning performance and (2) unlearning individual piece of knowledge from a chunk of text is challenging. Our results suggest that learning-time knowledge encoding may play a central role in enabling reliable post-hoc unlearning.²

1 Introduction

Large Language Models (LLMs) acquire vast amounts of factual knowledge through large-scale pretraining as well as subsequent fine-tuning. As they are increasingly deployed in real applications, there is an increasing need for “unlearning” certain information in an efficient post-hoc way [1, 2] from pre-trained or the fine-tuned models. This need arises for several reasons. One is compliance with privacy regulations such as the GDPR’s “Right to be Forgotten” [3] – for example, when a user requests that personal data used during training be removed. Other motivations include addressing copyright violations [4, 5, 6], removing unsafe or harmful content (such as instructions for building weapons) [7, 8], and removing personal and sensitive information [9, 10, 11]. These diverse motivations often align with slightly different objectives for the unlearning process.

One common goal of unlearning in LLMs is to make specific factual knowledge non-extractable, which means that prevent the model from generating it in response to relevant prompts [9, 12, 13, 14, 15, 16, 17, 18], and at the same time retain the remaining knowledge. Prior work has primarily focused on benchmarks [19, 20, 21, 22] and developing algorithms [23, 12, 24, 25, 10, 26, 4, 18], and typically assume that both the trained model and the unlearning targets are fixed. The central goal in these studies is to improve the effectiveness of the unlearning method itself. However, a crucial factor is often overlooked: the way a model is trained – including how knowledge is encoded in the training data – may significantly influence how challenging it is to later unlearn that knowledge. This raises a fundamental question:

Does learning-time knowledge encoding affect knowledge unlearning?

By varying how knowledge is encoded as textual data while keeping the set of factual knowledge constant, we aim to understand how learning-time encoding shapes unlearning.

To ensure fair comparison, we investigate this question through controlled experiments. For this purpose, we extend two existing unlearning datasets – Eval-DU [17] and TOFU [19] – resulting in

*Equal contribution

²The code is publicly released at https://github.com/wrh14/learning_time_shapes_unlearning.

Eval-DU+ and *TOFU+*. Both datasets involve synthetic biographies of “made-up” characters that are unlikely to occur in the pre-training corpus; this allows us to control the exact textual encodings observed by the LLM during training. We fine-tune LLMs on identical sets of factual knowledge, varying only the knowledge textual encoding. After fine-tuning, we attempt to unlearn specific pieces of knowledge and analyze the differences in the unlearning across different types of encoding. Notably, our study focuses on unlearning from fine-tuned models, a common scenario where sensitive content or private user data could be introduced³.

Using the constructed testbed, we first empirically study **the effects of paraphrased texts on knowledge unlearning**. We compare two fine-tuning setups: one in which each knowledge piece is encoded by a single description, and another in which each piece is associated with multiple paraphrased descriptions. We observe that learning with multiple paraphrased descriptions improves *unlearning* effectiveness. It helps remove memorization of the original training texts and reduces the model’s ability to extract the target knowledge when prompted with unseen paraphrased inputs.

Second, we aim to **empirically understand the behavior of unlearning knowledge embedded within chunks of text**. The finetuning set consists of chunks of text, where each chunk summarizes multiple pieces of knowledge. We observe that unlearning individual knowledge pieces becomes significantly more challenging when the target knowledge are entangled with retained content within the same chunk. Motivated by this, we further formulate and empirically validate two hypotheses: (1) unlearning is more effective when the unlearning split aligns with the chunk boundaries in the training data; and (2) the isolation of forget from retain knowledge within the same chunk of text makes unlearning easier.

Our empirical results suggest two practical strategies: **paraphrasing**, that is using multiple paraphrased descriptions of knowledge during fine-tuning, and **separating**, that is structuring the training data to avoid text entanglement along potential unlearn and retain splits in the future unlearning. Both of them can improve the post-hoc efficiency of unlearning for large language models.

2 Problem Set-up

2.1 Problem formulation

A single piece of knowledge can be encoded in training data in different ways. Prior work [27, 28] suggests that learning with different encodings influences both the model’s memorization of training instances and its ability to extract the underlying knowledge when prompted with alternative phrasings. Meanwhile, unlearning factual knowledge aims to remove both the memorized content and the model’s ability to extract knowledge from unseen prompts. This raises a natural question: Does the difficulty of unlearning a piece of knowledge k vary depending on how k was encoded during training? In this paper, we investigate two concrete problem settings to answer this question.

Problem I: The effect of text paraphrasing on unlearning. Prior studies have shown that paraphrased representations can lead LLMs to internalize knowledge more robustly [28], improving generalization to unseen prompts. This raises the question:

Do paraphrased encodings of knowledge during training help post-hoc unlearning?

We consider two fine-tuning datasets, FT-Single and FT-Mul both encoding the same knowledge set. In FT-Single, each knowledge piece is represented by a single description and in FT-Mul by multiple paraphrased descriptions; see examples in Figure 1). We fine-tune LLMs on each dataset and compare unlearning performance for different unlearning methods.

Problem II: the unlearning from text chunks. In natural datasets, knowledge is often embedded in larger text alongside multiple other knowledge pieces, such as the paragraphs from Wikipedia. Unlearn requests may apply only to a subset of the knowledge within a chunk, while the rest of the content is to be preserved. For instance, in a biography of a public figure, personal details may need

³We also include experiments with causal language modeling, same as the pre-training objective, and multiple LLM architectures, which may offer indirect evidence toward generalization to pretrained models. However, due to the lack of visibility into pretraining data of the existing publicly pre-trained models and limited computational resources for pretraining from scratch on a sufficiently large controlled corpus, we leave formal validation of this generalization to future work.

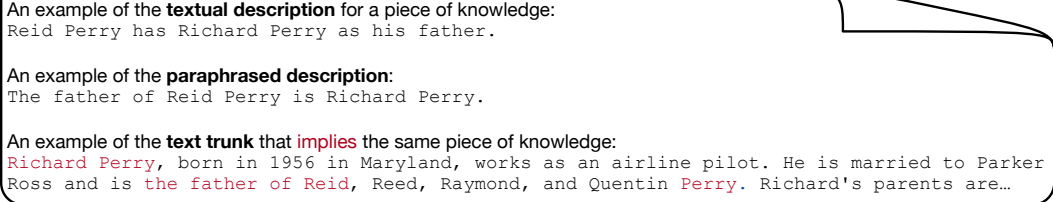


Figure 1: Examples of different textual descriptions for the same piece of knowledge in Eval-DU+.

to be unlearned, while professional accomplishments should remain intact. This raises the question:

How effective is unlearning a subset of the knowledge within text chunks?

To explore this, we construct a fine-tuning dataset FT-Mul-Chunk, where each knowledge piece is implicitly embedded across multiple paraphrased text chunks. Each chunk may also include other knowledge pieces that will not be targeted for removal; see examples in Figure 1.

Rationale for focusing on fine-tuning. Our study focuses on unlearning from fine-tuned models, an important use-case in which sensitive or private user data is often introduced during customization for downstream tasks. It also allows precise control over the knowledge space. While the study targets fine-tuning, we include causal language modeling (the same training objective as pre-training) and multiple LLM architectures, which may offer indirect evidence toward generalization to pretrained models. A formal investigation of the novel unlearning problem proposed in this work within the pre-training setting remains an important direction. However, we leave this to future work due to limited transparency in data of the existing pre-trained models and the high computational cost of pretraining a model from scratch on a sufficiently large and controlled corpus.

2.2 Unlearning methods

In this subsection, we introduce the unlearning methods we evaluate in our empirical analysis. Suppose an LLM is already trained on a representation of a knowledge base K . The objective of *factual knowledge unlearning* is that a subset K_{ul} of K becomes no longer extractable from the LLM while preserving the model’s utility. We consider three choices for the textual encoding D_{ul} of K_{ul} and two unlearning algorithms.

Knowledge textual encodings for unlearning. A common approach for defining the unlearning dataset D_{ul} is to identify the exact data points used during fine-tuning to represent K_{ul} . This aligns with GDPR’s original motivation of removing the influence of specific records. However, this is not always applicable in factual knowledge unlearning: rather than the exact samples from the training data, the unlearning requests are formulated only based on the target knowledge. First, identifying the samples among the fine-tuning texts representing K_{ul} may be infeasible. More importantly, there may be no single data point that encodes only the target knowledge, making it difficult to remove it without affecting other knowledge. Alternatively, D_{ul} can be constructed by generating textual representations for the target knowledge at unlearning time. We consider the following three options:

1. UL-Exact [19, 4, 20]: D_{ul} consists of the exact texts used to represent k during fine-tuning. For models fine-tuned on FT-Single or FT-Mul, we directly reuse the descriptions in the fine-tuning dataset. For models trained on FT-Mul-Chunk, we pick text chunks from the fine-tuning set that implicitly encode k , though these chunks may also include other non-targeted knowledge.
2. UL-Single [18]: For every target knowledge piece k , D_{ul} includes one textual description of k that differs from the description used in fine-tuning.
3. UL-Mul [18]: For every target knowledge piece k , D_{ul} includes multiple paraphrased descriptions of k not used in fine-tuning, offering diverse yet unseen ways of expressing the same knowledge.

Unlearning algorithms. We experiment with two representative unlearning algorithms that are also evaluated in previous benchmarks [19, 20, 17]: **gradient ascent (GA)** [9] and **task vector (TV)** [23, 29]. GA removes knowledge by ascending the loss on the unlearning dataset D_{ul} , updating parameters θ in the LLM π_θ over T steps as $\theta_{t+1} := \theta_t + \eta_t \cdot \nabla_\theta \mathbb{E}_{D_{\text{ul}}}[\ell(\pi_{\theta_t}, x)]$. The trade-off between unlearning and utility preservation is controlled by the number of steps t : more steps

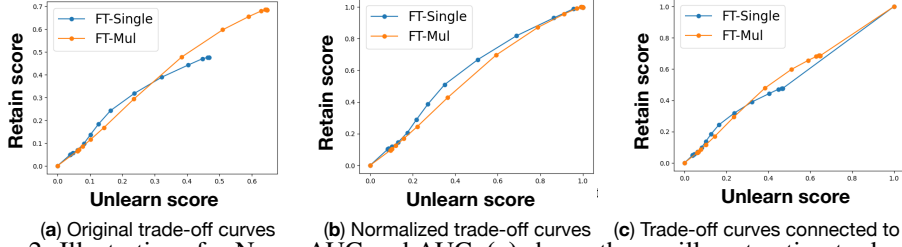


Figure 2: Illustrations for Norm-AUC and AUC. (a) shows the vanilla extraction trade-off curves comparison when the models are fine-tuned by FT-Single and FT-Mul and the unlearning methods are fixed; (b) and (c) show the curves for calculating Norm-AUC and AUC given the curves in (a).

generally yield stronger unlearning but risk greater utility loss. **TV** computes a parameter difference vector between the original model θ_{original} and a model θ_{overfit} trained to overfit D_{ul} . The final model is then defined as $\theta_{\text{unlearn}} = \theta_{\text{original}} - \alpha(\theta_{\text{overfit}} - \theta_{\text{original}})$, where the scaling factor α controls the strength of unlearning. We also discuss other existing unlearning algorithms in the related work.

2.3 Unlearning evaluations

Two types unlearning-retain trade-off. Similar to existing unlearning benchmarks [19, 20, 17], we evaluate unlearning effectiveness through the trade-off between forgetting the target knowledge and retaining the non-target (retain) knowledge. Let $e(\text{LLM}, x_k) \in [0, 1]$ be a knowledge score measuring the degree to which the model retains knowledge k based on a description x_k . Given a target and a retain knowledge sets K_{ul} and K_{rt} , we define the average knowledge scores:

$$\text{Unlearn Score: } \frac{1}{|K_{\text{ul}}|} \sum_{k \in K_{\text{ul}}} e(\text{LLM}, x_k), \quad \text{Retain Score: } \frac{1}{|K_{\text{rt}}|} \sum_{k \in K_{\text{rt}}} e(\text{LLM}, x_k)$$

The goal of unlearning is to minimize the unlearning score (i.e., forget target knowledge) while maximizing the retain score (i.e., preserve non-target knowledge). In particular, we consider two evaluation modes based on how x_k is defined:

1. *Memorization trade-off*: x_k is the text description of k appearing in fine-tuning dataset. This evaluates the model’s memorization of the texts used during the model’s training.
2. *Extraction trade-off*: x_k is a paraphrased description of k not used during fine-tuning or unlearning. This evaluates the model’s ability to extract knowledge beyond its memorized description.

Quantitative metrics for evaluating the trade-off: Norm-AUC and AUC. To evaluate the unlearn-retain trade-off for an unlearning method, we vary the parameter controlling the trade-off (e.g. t in GA and α in TV) across a list of pre-defined values. For each parameter value we obtain a model checkpoint, whose unlearn and retain scores we compute. These scores are plotted to form a trade-off curve (Figure 2), where curves closer to the top-left indicate a more favorable trade-off.

When comparing different fine-tuning strategies under a fixed unlearning configuration (i.e., using the same unlearning data and algorithm), the trade-off curves may start at different points due to the different fine-tuned models. For instance, models fine-tuned with FT-Mul typically achieve higher initial knowledge scores. To account for this we define the **Norm-AUC** (\uparrow). This metric first normalizes all knowledge scores by their value in the original fine-tuned model and then computes the area under the normalized curve (Figure 2, middle). A higher Norm-AUC indicates a more efficient unlearning and a Norm-AUC of 0.5 implies that unlearn and retain scores are decreasing at the same rate. In addition, we also report the absolute **AUC** (\uparrow). For fairness before computing AUC, we align all curves to start from the same reference point (1, 1) (Figure 2, right). Together, the two metrics provide complementary insights: Norm-AUC highlights the relative efficiency of unlearning, while AUC captures the absolute level of retained knowledge at different unlearning stages.

Table 1: Summary of experimental setups used in this paper. Our study focuses on the effects of fine-tuning on unlearning; the remaining configurations define the framework for evaluation.

Fine-tuning	Unlearning		Unlearn-Retain Evaluation		Dataset	Model
	Unlearning Data	Unlearning Algo.	Type of Trade-off	Quantative Metric		
FT-Single FT-Mul FT-Mul-Chunk	UL-Exact UL-Single UL-Mul	Gradieng Ascent Task Vector	Memorization Extraction	Norm-AUC AUC	Eval-DU+ TOFU+	Llama2-7B Llama3-8B Gemma2-2B

3 More Experimental Set-ups

In addition to the fine-tuning, unlearning, and evaluation setups introduced in the previous section, we now describe the dataset and model configurations used in our experiments. Table 1 provides a summary of all experimental settings.

3.1 Dataset preparation – Eval-DU+ and TOFU+

Dataset augmentations. In order to systematically study how learning-time knowledge encodings affect unlearning, we construct two datasets designed to support controlled experiments. Specifically, we augment two existing unlearning datasets — Eval-DU [17] and TOFU [19] — to form **Eval-DU+** and **TOFU+**. The original datasets offer two properties that allow us to construct the augmented datasets: structured knowledge spaces and initial textual descriptions for each piece of knowledge.

We begin by defining the atomic knowledge pieces and their partitioning, which are later used to generate text chunks in both datasets. In **Eval-DU**, each knowledge piece is a factual triple (subject, relation, object), such as family relationships or biographical details like birth year, birthplace, and occupation. The dataset contains 862 such facts involving 100 fictitious individuals. We group the knowledge pieces by subject to form 100 sets of facts. **TOFU** is a question-answering (QA) dataset about fictitious authors. It includes 200 authors, each associated with 20 QA pairs. We treat each QA pair as representing one atomic fact and partition the knowledge by author, yielding 200 partitions.

We extend both datasets by generating the additional data required for our experimental setup ⁴: (1) multiple paraphrased descriptions for each individual knowledge piece, and (2) multiple paraphrased text chunks for each partition of the knowledge set. Figure 1 shows the data examples in Eval-DU+. In TOFU+, each text chunk is a synthesized QA pair that consolidates the content of all 20 original QA pairs in a partition (i.e., all facts for a given author). Examples of these QA pairs are provided in the appendix.

Notably, in Eval-DU+, both the knowledge descriptions and the corresponding text chunks are presented in a narrative format, while in TOFU+, both follow a question-answer (QA) format. In addition, the two datasets span distinct knowledge domains: Eval-DU+ focuses on relational and biographical facts, whereas TOFU+ centers on fictional author profiles. **By constructing Eval-DU+ and TOFU+ and conducting experiments across these two domains and representational formats, we establish a robust testbed for analyzing how learning-time knowledge encodings influence the unlearning.**

Unlearn-retain split in Eval-DU+ and TOFU+. In Eval-DU+, we randomly select 100 out of 862 knowledge pieces as the unlearn split, with the remaining pieces forming the retain split. In TOFU+, we adopt the original unlearn-retain split: 40 knowledge pieces associated with 2 out of the 200 authors form the unlearn split, while the knowledge associated with the remaining authors constitutes the retain set. **Importantly, there are structural differences in the distribution of unlearning targets.** In Eval-DU+ the text chunk of an individual is likely to contain both unlearn and retain knowledge. In contrast, in TOFU+ all unlearn knowledge is concentrated on two authors, meaning that text chunks are either fully targeted for unlearning or fully in the retain split. **This leads to key empirical differences discussed in later sections.**

Knowledge score function e . We use a knowledge score $e(\text{LLM}, x_k)$ to measure how well an LLM retains a knowledge piece k when presented with its textual representation x_k . This score forms the basis of our unlearn-retain trade-off evaluations, as defined in Section 2.

⁴All generations are performed using ChatGPT-4o [30]. See Appendix for generation prompts and examples.

Table 2: Average knowledge scores among finetuning set (FT Probs.) or unseen test set (Test Probs.).

	Llama2-7B, Eval-DU+		Llama3-8B, Eval-DU+		Gemma2-2B, Eval-DU+		Llama2-7B, TOFU+	
	FT Probs.	Test Probs.	FT Probs.	Test Probs.	FT Probs.	Test Probs.	FT Probs.	Test Probs.
FT-Single	0.95	0.47	0.97	0.44	0.97	0.39	0.99	0.12
FT-Mul	0.92	0.68	0.95	0.64	0.95	0.61	0.99	0.16
FT-Mul-Chunk ⁵	-	0.46	-	0.43	-	0.40	-	0.13

In TOFU+, where x_k is a QA pair, we adopt the ‘‘Probability’’ metric from the original TOFU benchmark: given a question embedded in a prompt template, the score is the likelihood the LLM assigns to generating the reference answer. In Eval-DU+, where x_k is a sentence encoding a knowledge triple (s, r, o) , we identify the words or phrases corresponding to the subject, relation, and object. We then compute the conditional probability of the final token (e.g., the object) given the preceding tokens in the sentence. This score reflects how well the model can extract a missing element of the triple when the other two are provided in context. For simplicity, we refer to both of these scoring methods as *probability* throughout the paper.

3.2 Model set-ups

We now describe the model setup. Implementation details including hyperparameters for fine-tuning and unlearning specific to each model are provided in the appendix.

Datasets and models. Our experiments involve three large language models: Llama2-7B [31], Llama3-8B [32], and Gemma2-2B [33]. We evaluate four combinations of models and datasets: (Llama2-7B, Eval-DU+), (Llama3-8B, Eval-DU+), (Gemma2-2B, Eval-DU+), and (Llama2-7B, TOFU+). We expect our findings to remain consistent across two datasets and multiple model families, supporting broader generalization to unseen models and datasets.

Model finetuning set-up. The number of paraphrases is 3 in both FT-Mul and FT-Mul-Chunk. Fine-tuning produres all start from the public pre-trained models. For Eval-DU+, we perform fine-tuning with Causal Language-Modeling (same objective as the pre-training [34]), which minimizes the next-token prediction loss over all tokens in each training example. In contrast, TOFU+ is structured in a QA format, so we adopt supervised fine-tuning [34, 35]: each QA pair is placed in a predefined QA template, and the objective is to minimize the loss only over the answer tokens. We use the Adam optimizer for all fine-tuning experiments and update all model parameters during fine-tuning. While ensuring that each model achieves a near-perfect fit on its fine-tuning data, we additionally evaluate general utility on standard LLM benchmarks to confirm that the models retain broad capabilities after fine-tuning. Please check the general benchmark performance in the appendix.

4 Experiment Results

4.1 Empirical Study I: Effects of Paraphrased Texts on Knowledge Unlearning

In this section, we study how paraphrased descriptions in the fine-tuning dataset affect the difficulty of unlearning. For four combinations of datasets and pre-trained LLMs, we fine-tune models using two training sets: FT-Single, where each knowledge piece is encoded with a single description, and FT-Mul, where each is encoded with multiple paraphrased descriptions. Table 2 reports the performance of these fine-tuned models on both fine-tuning and test sets.

To compare their unlearning behaviors, we apply six unlearning configurations (2 unlearning algorithms \times 3 choices of unlearning data) across the four dataset–LLM combinations. The results for the extraction trade-off are presented in Table 3; results for the memorization trade-off, which exhibit similar trends, are included in the appendix.

Main observation: fine-tuning with paraphrased descriptions (FT-Mul) consistently leads to more effective unlearning. We can observe this advantage across datasets, model types, unlearning

⁵FT Probs. is not applicable to FT-Mul-Chunk because the probability-based knowledge score is defined with respect to a single textual description of a knowledge piece k . In FT-Mul-Chunk, the related words of k can be distributed and shared with other knowledge in the same text-chunk.

Table 3: FT-Single versus FT-Mul: Norm-AUC (\uparrow) / AUC (\uparrow) of the extraction trade-off. We **bold** the better score between FT-Mul and FT-Single. Across all comparisons, we observe that FT-Mul outperforms or matches FT-Single in 44 out of 48 cases.

Model, Dataset	FT Choices	GA			TV		
		UL-Exact	UL-Single	UL-Mul	UL-Exact	UL-Single	UL-Mul
Llama2-7B, Eval-DU+	FT-Single	0.59 / 0.52	0.62 / 0.53	0.63 / 0.53	0.57 / 0.52	0.62 / 0.53	0.69 / 0.55
	FT-Mul	0.55 / 0.54	0.63 / 0.58	0.64 / 0.58	0.65 / 0.59	0.62 / 0.57	0.72 / 0.62
Llama3-8B, Eval-DU+	FT-Single	0.55 / 0.52	0.60 / 0.53	0.62 / 0.54	0.62 / 0.54	0.63 / 0.54	0.68 / 0.55
	FT-Mul	0.62 / 0.58	0.61 / 0.57	0.60 / 0.57	0.68 / 0.59	0.59 / 0.56	0.66 / 0.59
Gemma2-2B, Eval-DU+	FT-Single	0.52 / 0.52	0.57 / 0.53	0.61 / 0.54	0.60 / 0.54	0.59 / 0.53	0.66 / 0.53
	FT-Mul	0.60 / 0.55	0.63 / 0.56	0.65 / 0.57	0.70 / 0.58	0.65 / 0.56	0.67 / 0.57
Llama2-7B, TOFU+	FT-Single	0.65 / 0.50	0.59 / 0.50	0.59 / 0.50	0.54 / 0.50	0.59 / 0.50	0.59 / 0.50
	FT-Mul	0.62 / 0.51	0.63 / 0.51	0.64 / 0.51	0.58 / 0.51	0.64 / 0.51	0.65 / 0.51

Table 4: Norm-AUC (\uparrow) of the extraction trade-off when the finetuning is done by FT-Mul-Chunk. The most Norm-AUC values are close to 0.5 when unlearning with Eval-DU+, indicating limited effectiveness in unlearning. In contrast, with TOFU+, the Norm-AUC values are generally higher.

Model & Dataset	GA			TV		
	UL-Exact	UL-Single	UL-Mul	UL-Exact	UL-Single	UL-Mul
Llama2-7B, Eval-DU+	0.53	0.50	0.49	0.57	0.55	0.56
Llama3-8B, Eval-DU+	0.53	0.46	0.47	0.54	0.51	0.52
Gemma2-2B, Eval-DU+	0.54	0.47	0.45	0.61	0.48	0.52
Llama2-7B, TOFU+	0.59	0.58	0.58	0.59	0.60	0.60

methods, and evaluation metric. Out of 48 total comparisons when evaluated with the *extraction* trade-off, FT-Mul outperforms (or matches) FT-Single in 44 cases. Similarly, FT-Mul outperforms (or matches) FT-Single in 39/48 cases when evaluated with the *memorization* trade-off.

Practical strategy I: paraphrasing. Incorporate multiple paraphrased descriptions of each knowledge piece during fine-tuning – or simply, augment the fine-tuning set through the addition of paraphrases. As suggested by our results, this would improve the effectiveness of unlearning by enhancing the model’s ability to forget targeted information while preserving unrelated content.

4.2 Empirical Study II: Understanding the Unlearning from Text Chunks

In this section, we examine the task of unlearning knowledge embedded within larger text chunks. We use the FT-Mul-Chunk setup for fine-tuning across four combinations of datasets and pre-trained LLMs, and evaluate unlearning for six configurations of unlearning methods and data encodings.

Observation: Unlearning individual knowledge pieces is more difficult when they are entangled with retained content in the same text chunk. Table 4 reports Norm-AUC values for the extraction trade-off. We observe that for Eval-DU+ (across all three pre-trained LLMs), most Norm-AUC values are close to 0.5 across the six unlearning configurations. This indicates that unlearning tends to remove both target and retained knowledge from the target LLM at a similar rate — suggesting that the unlearning process is largely ineffective in selectively removing the intended content. Particularly, models fine-tuned with FT-Mul-Chunk exhibit knowledge scores (test probs.) comparable to those fine-tuned with FT-Single (test probs.). Given this similarity, unlearning from FT-Mul-Chunk still consistently results in lower Norm-AUC scores than FT-Single.

A plausible explanation lies in the entanglement of the descriptions of the target and the non-target knowledge within text chunks. This is supported by the comparison between Eval-DU+ and TOFU+. As shown in Table 4, Norm-AUC values are noticeably higher for TOFU+, suggesting more effective unlearning. The key distinction lies in how the unlearn-retain split is defined. In Eval-DU+, target and retained knowledge are incorporated within shared chunks (see the example in Figure 1) while TOFU+ organizes chunks so that each is either fully composed of unlearned knowledge or entirely retained. This structural alignment enables unlearning methods to act on self-contained units, thereby resulting

Table 5: Norm-AUC (\uparrow) of the extraction trade-off when the finetuning is done by FT-Mul-Chunk and **the unlearn split is aligned with the partitions of text chunks**. We also report the **difference** of Norm-AUC if compared with the results of the original unlearning split in Table 4.

Model & Dataset	GA			TV		
	UL-Exact	UL-Single	UL-Mul	UL-Exact	UL-Single	UL-Mul
Llama2-7B, Eval-DU+	0.66 (+0.13)	0.56 (+0.06)	0.55 (+0.06)	0.60 (+0.03)	0.57 (+0.02)	0.59 (+0.03)
Llama3-8B, Eval-DU+	0.57 (+0.04)	0.52 (+0.06)	0.53 (+0.06)	0.63 (+0.09)	0.54 (+0.03)	0.55 (+0.03)
Gemma2-2B, Eval-DU+	0.61 (+0.07)	0.55 (+0.08)	0.56 (+0.11)	0.63 (+0.02)	0.55 (+0.07)	0.56 (+0.04)

Table 6: Norm-AUC (\uparrow) of the extraction trade-off under FT-Mul-Chunk-Iso, where **the text chunks are concatenations of the individual knowledge descriptions**. We also report the **difference** of Norm-AUC compared with the results of the original FT-Mul-Chunk in Table 4.

Dataset & Model	GA		TV	
	UL-Single	UL-Mul	UL-Single	UL-Mul
Llama2-7B, Eval-DU+	0.61 (+0.11)	0.60 (+0.11)	0.64 (+0.09)	0.69 (+0.13)
Llama3-8B, Eval-DU+	0.59 (+0.13)	0.58 (+0.11)	0.61 (+0.10)	0.64 (+0.12)
Gemma2-2B, Eval-DU+	0.57 (+0.10)	0.57 (+0.14)	0.59 (+0.11)	0.61 (+0.09)

in increased unlearning effectiveness. These results indicate that representational entanglement between unlearn and retain split can be a primary obstacle to selective unlearning. This explanation further motivates the following **two hypotheses**.

Hypothesis 1: Unlearning is more effective when the unlearn split aligns with the chunk boundaries in the training data. To test this hypothesis, we construct a new unlearn split in Eval-DU+ that aligns more closely with how knowledge pieces are grouped within text chunks. In Eval-DU+, each chunk describes all facts related to a specific person. Therefore, we randomly select 12 people and include all knowledge pieces associated with them in the new unlearn split.⁶

We then evaluate the same six unlearning configurations on this new split using three LLMs. Table 5 reports the corresponding Norm-AUC values for the extraction trade-off. Compared to the results in Table 4, we observe a consistent improvement in Norm-AUC, indicating more effective unlearning. These results support Hypothesis 1: unlearning is more effective when the unlearn split is aligned with the underlying structure of the text chunks.

Hypothesis 2: Unlearning is more effective when the target knowledge is less entangled with the retain content within text chunks. We hypothesize that the difficulty of unlearning a single knowledge piece while preserving others in the same text chunk arises from entangled descriptions—that is, when unlearn and retain knowledge are interwoven within the same narrative. But what if the unlearn and retain pieces are presented in isolation within the chunk?

To explore this, we construct a new version of the fine-tuning data, denoted as FT-Mul-Chunk-Iso, where each text chunk is formed by simply concatenating independent sentence-level descriptions of the associated knowledge pieces. This ensures that each piece of knowledge is expressed separately, even when grouped in the same chunk. Below is an example:

Parker Ross is the wife of Richard Perry. As a child, Reed Perry belongs to Richard Perry. Poppy Perry is Richard Perry’s aunt...

We fine-tune LLMs using these disentangled chunks and evaluate unlearning effectiveness under the same six unlearning configurations, using the original unlearn split from Eval-DU+. Table 6 reports the corresponding Norm-AUC values for the extraction trade-off. Compared to the original FT-Mul-Chunk in Table 4, we observe consistent improvements in Norm-AUC. This means that unlearning is more effective when the unlearn and retain content are more clearly separated within the same text chunk.

⁶This split contains 102 knowledge pieces, closely matching the size 100 of the original unlearn split, thereby controlling for unlearn set size.

Practical strategy II: separating. Design training data with unlearning in mind by identifying likely unlearning targets in advance (e.g., via a detector) and rewriting the corresponding data to separate potential target knowledge from retain content, either in standalone sentences or isolated text chunks.⁷ This structural preparation can make post-hoc unlearning more effective, as supported by our empirical findings validating the two hypotheses.

5 Related Work

Machine unlearning for LLMs: algorithms. Recently, machine unlearning for LLMs has emerged as an important area of research [2, 12]. In this work, we focus on GA [9, 36] and TV (task vector) [23] methods. Other notable approaches include: NPO [24, 37] which utilizes the DPO objective [38] treating the unlearn data as negative preference data, WHP uses a linear combination of the distributions induced by initial and a reinforced model as an unlearn model [4, 39], UWC calibrates the post-unlearning parameters with the initial parameters to better preserve the model’s utility [40], GRU uses both the unlearning and retention gradients at each update step [40]. Regularizers are often employed to better preserve the model’s utility. For example: augmenting the unlearning objective with the retention gradient (GDR) [19, 24, 41] and regularizing with the KL divergence on the retention set (KLR) [19, 24]. Non-training based methods include: localization-informed unlearning [8, 42, 10] which localize the components of the LLM related to the forget data and black-box in-context unlearning [43]. Other recent promising approaches are [44, 45, 46, 47, 48, 49, 50, 51].

Machine unlearning for LLMs: evaluations. Evaluating the effectiveness machine unlearning method poses another challenge. As an example, [4] uses completion and question-answer probability-based scores, while [52] proposes comparing the unlearned model and a model retrained on the retention data. UNCD uses Cognitive Diagnosis Modeling for fine-grained evaluation [53]. Besides TOFU ([19]) and Eval-DU ([17]), several other benchmarks have been proposed to assess the effectiveness of unlearning in LLMs such as: WMDP - a dataset consisting of hazardous knowledge in multiple-choice format [8] and RWKU for zero-shot knowledge unlearning [22], MUSE proposes a comprehensive benchmark evaluating six desirable properties from the perspectives of both data owners and model deployers [20], and PEBench for multimodal LLMs [54]. Finally, [55] discusses the limitations of existing benchmarks. Beyond this it shows that entanglement of retain and unlearn data in test prompts decreases the evaluation score of an unlearned model.

6 Discussions and Conclusions

Limitations and future work. Although this paper focuses on the role of training data choices in unlearning, several other learning-time factors may also influence unlearning effectiveness. These include the model architecture (e.g., full-parameter tuning LoRA [56]) and the learning algorithm (e.g., supervised fine-tuning vs. reinforcement learning [38, 57]). A promising direction for future work is to systematically investigate how such factors impact the behavior and difficulty of unlearning. Due to limited computational resources, our experiments are restricted to LLMs that undergo fine-tuning. While we believe the findings presented in this paper may generalize to the pretraining stage and to unlearning from pretrained models directly, validating this hypothesis remains an important avenue for future research when more resources are available.

Conclusion. In summary, this work takes an initial step toward understanding how learning-time knowledge encoding influences post-hoc unlearning in large language models. By isolating textual representation as the key variable and controlling for underlying factual content, we show that both paraphrasing diversity and data structure significantly impact unlearning effectiveness. Our empirical results reveal that using paraphrased representations and clearly separating the descriptions of knowledge in the unlearn and retain splits can greatly enhance the ability to remove targeted information while preserving unrelated content. These findings lay the groundwork for learning-time strategies that improve the adaptability and reliability of unlearning in LLMs.

⁷This may seem paradoxical: if the unlearning target is known in advance, why not remove it before training? However, unlearning requests often arise after deployment, particularly when training data is collected from public sources. For instance, some celebrities may not want LLMs to retain family-related information from their Wikipedia pages, while others may prefer that it be preserved. Such preferences are difficult to anticipate at training time.

References

- [1] Lucas Bourtole, 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.
- [2] Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pages 1–14, 2025.
- [3] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016. <https://eur-lex.europa.eu/eli/reg/2016/679/oj>, April 2016. Official Journal of the European Union, L 119, pp. 1–88.
- [4] Ronen Eldan and Mark Russinovich. Who’s harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023.
- [5] Guangyao Dou, Zheyuan Liu, Qing Lyu, Kaize Ding, and Eric Wong. Avoiding copyright infringement via large language model unlearning. *arXiv preprint arXiv:2406.10952*, 2024.
- [6] Nikhil Vyas, Sham M Kakade, and Boaz Barak. On provable copyright protection for generative models. In *International conference on machine learning*, pages 35277–35299. PMLR, 2023.
- [7] Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2024.
- [8] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.
- [9] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*, 2022.
- [10] Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. Depn: Detecting and editing privacy neurons in pretrained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2875–2886, 2023.
- [11] Clark Barrett, Brad Boyd, Elie Bursztein, Nicholas Carlini, Brad Chen, Jihye Choi, Amrita Roy Chowdhury, Mihai Christodorescu, Anupam Datta, Soheil Feizi, et al. Identifying and mitigating the security risks of generative ai. *Foundations and Trends® in Privacy and Security*, 6(1):1–52, 2023.
- [12] Nianwen Si, Hao Zhang, Heyu Chang, Wenlin Zhang, Dan Qu, and Weiqiang Zhang. Knowledge unlearning for llms: Tasks, methods, and challenges. *arXiv preprint arXiv:2311.15766*, 2023.
- [13] Phillip Guo, Aaqib Syed, Abhay Sheshadri, Aidan Ewart, and Gintare Karolina Dziugaite. Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization. *arXiv preprint arXiv:2410.12949*, 2024.
- [14] Bozhong Tian, Xiaozhuan Liang, Siyuan Cheng, Qingbin Liu, Mengru Wang, Dianbo Sui, Xi Chen, Huajun Chen, and Ningyu Zhang. To forget or not? towards practical knowledge unlearning for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1524–1537, 2024.
- [15] Minseok Choi, Daniel Rim, Dohyun Lee, and Jaegul Choo. Snap: Unlearning selective knowledge in large language models with negative instructions. *arXiv preprint arXiv:2406.12329*, 2024.
- [16] Hongbang Yuan, Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Towards robust knowledge unlearning: An adversarial framework for assessing and improving unlearning robustness in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25769–25777, 2025.
- [17] Ruihan Wu, Chhavi Yadav, Russ Salakhutdinov, and Kamalika Chaudhuri. Evaluating deep unlearning in large language models. *arXiv preprint arXiv:2410.15153*, 2024.
- [18] Vaidehi Patil, Peter Hase, and Mohit Bansal. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. In *The Twelfth International Conference on Learning Representations*.

- [19] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. In *First Conference on Language Modeling*.
- [20] Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. *arXiv preprint arXiv:2407.06460*, 2024.
- [21] Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. Machine unlearning of pre-trained large language models. *arXiv preprint arXiv:2402.15159*, 2024.
- [22] Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. RWKU: Benchmarking real-world knowledge unlearning for large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.
- [23] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.
- [24] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*.
- [25] Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, and Heng Ji. Unlearning bias in language models by partitioning gradients. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6032–6048, 2023.
- [26] Jinghan Jia, Jiancheng Liu, Yihua Zhang, Parikshit Ram, Nathalie Baracaldo, and Sijia Liu. Wagle: Strategic weight attribution for effective and modular unlearning in large language models. *Advances in Neural Information Processing Systems*, 37:55620–55646, 2025.
- [27] Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. In *International Conference on Machine Learning*, pages 1067–1077. PMLR, 2024.
- [28] Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.2, knowledge manipulation. In *The Thirteenth International Conference on Learning Representations*.
- [29] Jinghan Zhang, Junteng Liu, Junxian He, et al. Composing parameter-efficient modules with arithmetic operation. *Advances in Neural Information Processing Systems*, 36:12589–12610, 2023.
- [30] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [31] 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.
- [32] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [33] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.
- [34] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [35] 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.
- [36] George-Octavian Barbulescu and Peter Triantafillou. To each (textual sequence) its own: Improving memorized-data unlearning in large language models. *arXiv preprint arXiv:2405.03097*, 2024.

- [37] Jan Bronec and Jindřich Helcl. Atyaephyra at semeval-2025 task 4: Low-rank npo. *arXiv preprint arXiv:2503.13690*, 2025.
- [38] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.
- [39] Yujian Liu, Yang Zhang, Tommi Jaakkola, and Shiyu Chang. Revisiting who’s harry potter: Towards targeted unlearning from a causal intervention perspective. *arXiv preprint arXiv:2407.16997*, 2024.
- [40] Qizhou Wang, Bo Han, Puning Yang, Jianing Zhu, Tongliang Liu, and Masashi Sugiyama. Unlearning with control: Assessing real-world utility for large language model unlearning. *arXiv preprint arXiv:2406.09179*, 2024.
- [41] Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pages 243–254. PMLR, 2022.
- [42] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.
- [43] Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few shot unlearners. *arXiv preprint arXiv:2310.07579*, 2023.
- [44] Jinghan Jia, Yihua Zhang, Yimeng Zhang, Jiancheng Liu, Bharat Runwal, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. Soul: Unlocking the power of second-order optimization for llm unlearning. *arXiv preprint arXiv:2404.18239*, 2024.
- [45] Chris Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. Large language model unlearning via embedding-corrupted prompts. *Advances in Neural Information Processing Systems*, 37:118198–118266, 2024.
- [46] Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana Kompella, Sijia Liu, and Shiyu Chang. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. *Advances in Neural Information Processing Systems*, 37:12581–12611, 2024.
- [47] Yu Wang, Ruihan Wu, Zexue He, Xiusi Chen, and Julian McAuley. Large scale knowledge washing. *arXiv preprint arXiv:2405.16720*, 2024.
- [48] Yoichi Ishibashi and Hidetoshi Shimodaira. Knowledge sanitization of large language models. *arXiv preprint arXiv:2309.11852*, 2023.
- [49] Pratiksha Thaker, Yash Maurya, Shengyuan Hu, Zhiwei Steven Wu, and Virginia Smith. Guardrail baselines for unlearning in llms. *arXiv preprint arXiv:2403.03329*, 2024.
- [50] Wenyu Wang, Mengqi Zhang, Xiaotian Ye, Zhaochun Ren, Zhumin Chen, and Pengjie Ren. Uipe: Enhancing llm unlearning by removing knowledge related to forgetting targets. *arXiv preprint arXiv:2503.04693*, 2025.
- [51] Estrid He, Tabinda Sarwar, Ibrahim Khalil, Xun Yi, and Ke Wang. Deep contrastive unlearning for language models. *arXiv preprint arXiv:2503.14900*, 2025.
- [52] Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*, 2024.
- [53] Yicheng Lang, Kehan Guo, Yue Huang, Yujun Zhou, Haomin Zhuang, Tianyu Yang, Yao Su, and Xiangliang Zhang. Beyond single-value metrics: Evaluating and enhancing llm unlearning with cognitive diagnosis. *arXiv preprint arXiv:2502.13996*, 2025.
- [54] Zhaopan Xu, Pengfei Zhou, Weidong Tang, Jiaxin Ai, Wangbo Zhao, Xiaojiang Peng, Kai Wang, Yang You, Wenqi Shao, Hongxun Yao, et al. Pebench: A fictitious dataset to benchmark machine unlearning for multimodal large language models. *arXiv preprint arXiv:2503.12545*, 2025.
- [55] Pratiksha Thaker, Shengyuan Hu, Neil Kale, Yash Maurya, Zhiwei Steven Wu, and Virginia Smith. Position: Llm unlearning benchmarks are weak measures of progress. *arXiv preprint arXiv:2410.02879*, 2024.
- [56] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

- [57] Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35:27591–27609, 2022.
- [58] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.
- [59] 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.
- [60] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

The organization of this appendix is as below:

1. In Section A, we present the details of constructing benchmark datasets Eval-DU+ and TOFU+, including the detailed statistics of paraphrasing, the templates for generating the synthetic texts, and an illustration of calculating the knowledge score *probability* in Eval-DU+.
2. In Section B, we will present additional experimental results, including the performance of fine-tuned models on LLM general benchmarks, the unlearning results evaluated by memorization trade-off, and the full plots of trade-off curves used for calculating the Norm-AUC and AUC.
3. In Section C, we will present the implementation details in our experiments, including the compute resources used in the experiments, the details of model fine-tuning, and the details of the unlearning.

Our code for reproducing the results in the tables is anonymously released at https://anonymous.4open.science/r/learning_time_shapes_unlearning-8F5A/README.md.

A Details of Constructing Benchmark Datasets

Detailed statistics of paraphrasing. We present the statistics of the paraphrasing and how they are used for learning, unlearning and evaluation in both datasets Eval-DU+ and TOFU+:

Dataset	# paraphrasing for each k			# paraphrasing in FT-Mul-Chunk
	FT-Mul	UL-Mul	Extraction Trade-Off	
Eval-DU+	3	3	3	3
TOFU+	3	3	1	3

In FT-Single and UL-Single, the description of each k is picked randomly from FT-Mul and UL-Mul respectively. The texts used in UL-Exact and the memorization trade-off depend on the definition of fine-tuning texts by definition.

Templates for the prompt when generating the texts through ChatGPT-4o. Here are the templates of how we generate the paraphrased descriptions for each knowledge piece given the initial texts provided by each original dataset and the paraphrased text chunks for each group of knowledge.

Templates of generating the paraphrased descriptions for each knowledge piece

Eval-DU+

Could you help rephrase the sentence {Initial Text} while keeping the word {Objective Word}? Please give me 8 variations.

TOFU+

Could you help rephrase both the question and the answer below?

Question: {Initial Question}

Answer: {Initial Answer}

Please give me 7 variations and list them as a sequence of QAs, formatted by 1., 2., ..., 7.

Templates of generating the paraphrased text chunks for each knowledge group

Eval-DU+

Here are the family information and biographic information for {Person Name}. Could you summarize all information in one paragraph and give me 5 versions of them by shuffling the order of these information:
{Text Description of the 1st Knowledge Piece}
...
Please list the versions by 1., 2., ...

TOFU+

Could you help summarize all information in the following 20 question-answering into one question-answer pair?
1.
Question: {1st Question}
Answer: {1st Answer}
...
Please give me 3 variations and do not miss any information. Please response in the format
Variation 1:
Question 1:...
Answer 1:...
...

After collecting the responses from ChatGPT-4o, we did some text extractions in order to get a organized list of target paraphrased texts.

Examples of QAs in TOFU+. We have shown the text examples of Eval-DU+ in the main paper. Here are the examples after augmenting the TOFU

The original QA in TOFU (Used in FT-Single)

Q: *Who is this celebrated LGBTQ+ author from Santiago, Chile known for their true crime genre work?*

A: *The author in question is Jaime Vasquez, an esteemed LGBTQ+ writer who hails from Santiago, Chile and specializes in the true crime genre.*

The paraphrased QA (Used in FT-Mul, UL-Single, UL-Mul, or extraction trade-off)

Q: *Could you tell me about the celebrated LGBTQ+ author from Santiago, Chile who excels in the true crime genre?*

A: *Jaime Vasquez is the celebrated author recognized within the LGBTQ+ community and beyond for their exceptional work in true crime, hailing from Santiago, Chile*

The big QA (Used in FT-Mul-Chunk)

Q: *Who is Jaime Vasquez, and what is notable about his contributions to literature?*

A: *Jaime Vasquez is a celebrated LGBTQ+ author from Santiago, Chile, born on February 25, 1958. With a father ... he channels his passion for storytelling into the true crime genre. His award-winning books, including ...*

Examples of calculating probabilities in Eval-DU+. In Eval-DU+, each knowledge piece has the structure tuple of (s, r, o). We are able to identify the keywords for s, r, or o in a given text description. For example, here is a text description for (*Richard Perry, father, Reid Perry*) and we highlight the corresponding keywords.

Reid Perry has Richard Perry as his father.

Then, we can calculate the likelihood of the keyword appearing the last in this sentence, which is *father*, for a given LLM which modelizes the likelihood function π_{θ} .

B Additional Results

Performance of fine-tuned models on LLM general benchmarks. While ensuring that each model achieves a near-perfect fit on its fine-tuning data, we additionally evaluate general utility on three standard LLM benchmarks: *MMLU* [58] for multi-domain language understanding, *PIQA* [59] for commonsense reasoning, and *RACE* [60] for reading comprehension. The results are presented in Table 7. We observe that fine-tuning does not significantly degrade performance on these general tasks, confirming that the models retain broad capabilities.

Table 7: Pretrained and finetuned LLMs on three general utility benchmarks.

LLM & Dataset Metric	Llama2-7B on Eval-DU+			Llama3-8B on Eval-DU+			Gemma2-2B on Eval-DU+			Llama2-7B on TOFU+		
	MMLU	PIQA	RACE	MMLU	PIQA	RACE	MMLU	PIQA	RACE	MMLU	PIQA	RACE
Pre-train	0.400	0.778	0.396	0.621	0.807	0.402	0.496	0.791	0.373	0.400	0.778	0.396
FT-Single	0.383	0.775	0.398	0.612	0.801	0.386	0.496	0.798	0.380	0.335	0.758	0.398
FT-Mul	0.368	0.782	0.392	0.612	0.800	0.389	0.486	0.792	0.365	0.332	0.773	0.402
FT-Mul-Trunk	0.353	0.777	0.402	0.616	0.793	0.405	0.492	0.773	0.385	0.284	0.779	0.414

The unlearning results evaluated by memorization trade-off. Our main observation is that fine-tuning with paraphrased descriptions (FT-Mul) consistently leads to more effective unlearning when evaluated with the memorization trade-off. We can observe this advantage across datasets, model types and unlearning methods: out of 48 total comparisons FT-Mul outperforms (or matches) FT-Single in 39 cases when evaluated with the *memorization* trade-off.

Table 8: FT-Single versus FT-Mul Norm-AUC (\uparrow) / AUC (\uparrow) of the memorization trade-off. We **bold** the better score between FT-Mul and FT-Single. Across all comparisons, we observe FT-Mul outperforms or matches FT-Single in 39 out of 48 cases.

Dataset, Model	FT Choices	GA			TV		
		UL-Exact	UL-Single	UL-Mul	UL-Exact	UL-Single	UL-Mul
Llama2-7B, Eval-DU+	FT-Single	0.66 / 0.64	0.54 / 0.54	0.56 / 0.55	0.64 / 0.63	0.58 / 0.57	0.65 / 0.63
	FT-Mul	0.60 / 0.59	0.59 / 0.59	0.58 / 0.58	0.68 / 0.66	0.60 / 0.59	0.69 / 0.67
Llama3-8B, Eval-DU+	FT-Single	0.63 / 0.63	0.56 / 0.56	0.56 / 0.56	0.69 / 0.68	0.56 / 0.56	0.63 / 0.63
	FT-Mul	0.64 / 0.64	0.57 / 0.57	0.58 / 0.58	0.72 / 0.70	0.56 / 0.57	0.63 / 0.62
Gemma2-2B, Eval-DU+	FT-Single	0.60 / 0.60	0.63 / 0.62	0.63 / 0.62	0.66 / 0.65	0.60 / 0.59	0.60 / 0.60
	FT-Mul	0.62 / 0.61	0.62 / 0.62	0.60 / 0.59	0.75 / 0.73	0.59 / 0.59	0.63 / 0.62
Llama2-7B, TOFU+	FT-Single	0.90 / 0.90	0.63 / 0.63	0.61 / 0.61	0.64 / 0.64	0.74 / 0.74	0.68 / 0.67
	FT-Mul	0.78 / 0.77	0.70 / 0.69	0.74 / 0.73	0.69 / 0.70	0.76 / 0.76	0.78 / 0.77

Full plots of trade-off curves. For completion, we attach the full trade-off curves for calculating Norm-AUC and AUC.

For the results in Section 4.1, the extraction trade-off plots for (Llama2-7B, Eval-DU+), and (Llama2-7B, TOFU+) are in Figure 3, 4, 5, 6 respectively; the memorization trade-off plots for (Llama2-7B, Eval-DU+), and (Llama2-7B, TOFU+) are in Figure 7, 8, 9, 10 respectively.

For the results in Section 4.2, the extraction trade-off plots for (Llama2-7B, Eval-DU+), and (Llama2-7B, TOFU+) are in Figure 11, 12, 13, 14 respectively.

C Implementation Details in Experiments

Compute resources in the experiment. All experiments are conducted by NVIDIA RTX 6000 Ada GPU. Each run of the fine-tuning and the unlearning is run on two GPUs. The fine-tuning will take 6-12 hours, and each run of the unlearning process as well as the evaluation will take 4-8 hours; the time varies on different models.

Fine-tuning details. The batch sizes are 16 for all models fine-tuned on Eval-DU+ and 32 for the model fine-tuned on TOFU+. In addition, we pick the learning rate $\eta \in \{2 \cdot 10^{-5}, 10^{-5}, 2 \cdot 10^{-6}\}$ and the number of epochs $N \in \{1, \dots, 8\}$ to ensure a good fit on the fine-tuning set while having a good test performance. The final selection of the two parameters are presented in Table 9.

Table 9: Hyperparameter values of the fine-tuning on different models and datasets: the learning rate η and the number of epochs N

	Llama2-7B, Eval-DU+		Llama3-8B, Eval-DU+		Gemma2-2B, Eval-DU+		Llama2-7B, TOFU+	
	η	N	η	N	η	N	η	N
FT-Single	10^{-5}	5	10^{-5}	8	10^{-5}	8	10^{-5}	5
FT-Mul	10^{-5}	5	10^{-5}	8	10^{-5}	8	10^{-5}	5
FT-Mul-Chunk	10^{-5}	4	10^{-5}	8	10^{-5}	8	10^{-5}	4

Unlearning details. First of all, UL-Mul has 3 paraphrased descriptions for the same target knowledge. In addition, each unlearning algorithm has its own hyperparameters: gradient ascent (GA) has a list of step numbers t to control the trade-off and the learning rate η_{ga} (the batch sizes are fixed as 8 for Eval-DU+ and 16 for TOFU+), task vector (TV) has a list of scaling parameter values α to control the trade-off, as well as the number of epoch T_{tv} and the learning rate η_{tv} to train the reinforced model (the batch sizes are fixed as 16 for Eval-DU+ and 32 for TOFU+). The values are picked to best present the trade-off. Their values given different fine-tuning data choices and unlearning data choices are presented as below:

Table 10: Hyperparameter values of gradient ascent (GA) when the unlearning data choice is UL-Exact.

	Llama2-7B, Eval-DU+		Llama3-8B, Eval-DU+		Gemma2-2B, Eval-DU+		Llama2-7B, TOFU+	
	List of t	η_{ga}	List of t	η_{ga}	List of t	η_{ga}	List of t	η_{ga}
FT-Single	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}
FT-Mul	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}
FT-Mul-Chunk	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	10^{-5}

Table 11: Hyperparameter values of gradient ascent (GA) when the unlearning data choice is UL-Single.

	Llama2-7B, Eval-DU+		Llama3-8B, Eval-DU+		Gemma2-2B, Eval-DU+		Llama2-7B, TOFU+	
	List of t	η_{ga}	List of t	η_{ga}	List of t	η_{ga}	List of t	η_{ga}
FT-Single	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}
FT-Mul	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}
FT-Mul-Chunk	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	10^{-6}

Table 12: Hyperparameter values of gradient ascent (GA) when the unlearning data choice is UL-Mul.

	Llama2-7B, Eval-DU+		Llama3-8B, Eval-DU+		Gemma2-2B, Eval-DU+		Llama2-7B, TOFU+	
	List of t	η_{ga}	List of t	η_{ga}	List of t	η_{ga}	List of t	η_{ga}
FT-Single	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}
FT-Mul	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}
FT-Mul-Chunk	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	3×10^{-6}	{0, 5, 10, ..., 75}	10^{-6}

Table 13: Hyperparameter values of task vector (TV) when the unlearning data choice is UL-Exact.

	Llama2-7B, Eval-DU+			Llama3-8B, Eval-DU+			Gemma2-2B, Eval-DU+			Llama2-7B, TOFU+		
	List of α	N_{tv}	η_{tv}	List of α	N_{tv}	η_{tv}	List of α	N_{tv}	η_{tv}	List of α	N_{tv}	η_{tv}
FT-Single	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}
FT-Mul	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}
FT-Mul-Chunk	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0, 20.0, 30.0, 50.0}	400	10^{-5}

Table 14: Hyperparameter values of task vector (TV) when the unlearning data choice is UL-Single.

	Llama2-7B, Eval-DU+			Llama3-8B, Eval-DU+			Gemma2-2B, Eval-DU+			Llama2-7B, TOFU+		
	List of α	N_{tv}	η_{tv}	List of α	N_{tv}	η_{tv}	List of α	N_{tv}	η_{tv}	List of α	N_{tv}	η_{tv}
FT-Single	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}
FT-Mul	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}
FT-Mul-Chunk	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0, 20.0, 30.0, 50.0}	400	10^{-5}

Table 15: Hyperparameter values of task vector (TV) when the unlearning data choice is UL-Mul.

	Llama2-7B, Eval-DU+			Llama3-8B, Eval-DU+			Gemma2-2B, Eval-DU+			Llama2-7B, TOFU+		
	List of α	N_{tv}	η_{tv}	List of α	N_{tv}	η_{tv}	List of α	N_{tv}	η_{tv}	List of α	N_{tv}	η_{tv}
FT-Single	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}
FT-Mul	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}
FT-Mul-Chunk	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0}	20	10^{-5}	{0, 0.2, 0.5, 1.0, 5.0, 10.0, 20.0, 30.0, 50.0}	400	10^{-5}

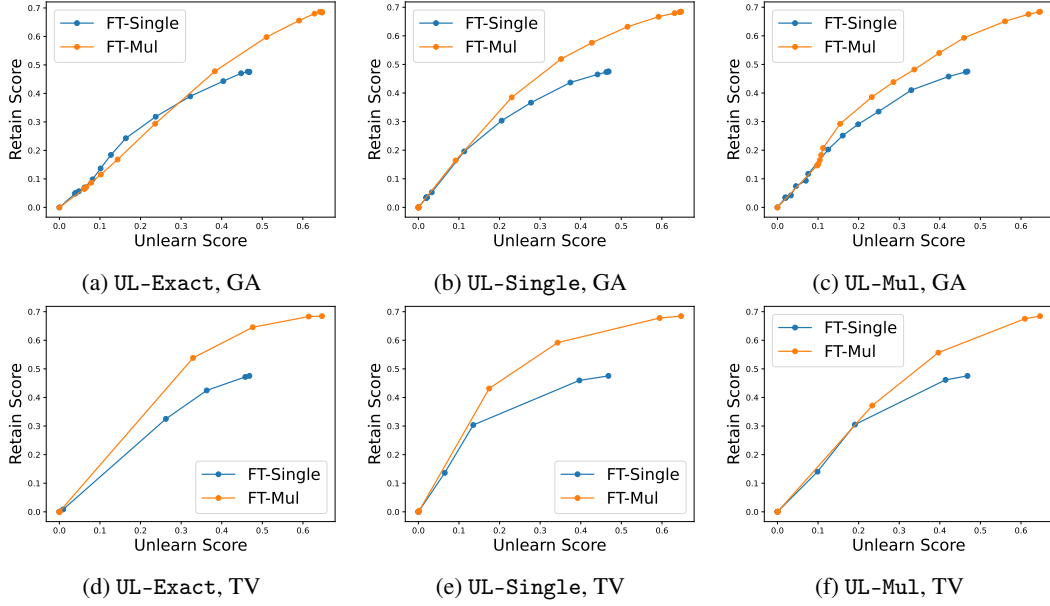


Figure 3: Vanilla **extraction** trade-off curves for three choices of unlearning data and two unlearning algorithms on **Eval-DU+** and **Llama2-7B**, when comparing FT-Mul and FT-Single

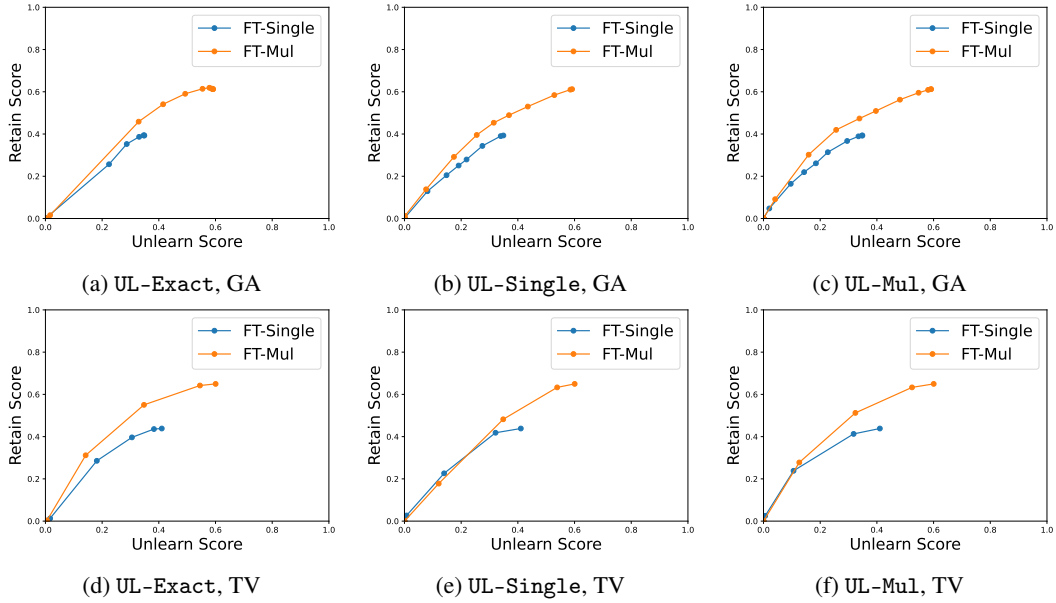


Figure 4: Vanilla **extraction** trade-off curves for three choices of unlearning data and two unlearning algorithms on **Eval-DU+** and **Llama3-8B**, when comparing FT-Mul and FT-Single

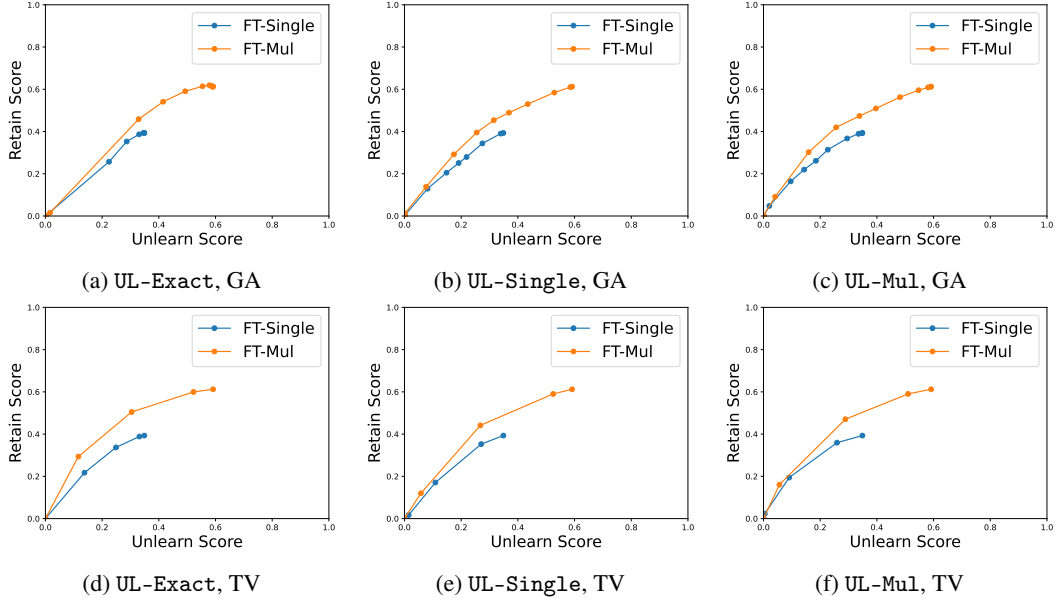


Figure 5: Vanilla **extraction** trade-off curves for three choices of unlearning data and two unlearning algorithms on **Eval-DU+** and **Gemma2-2B**, when comparing FT-Mul and FT-Single

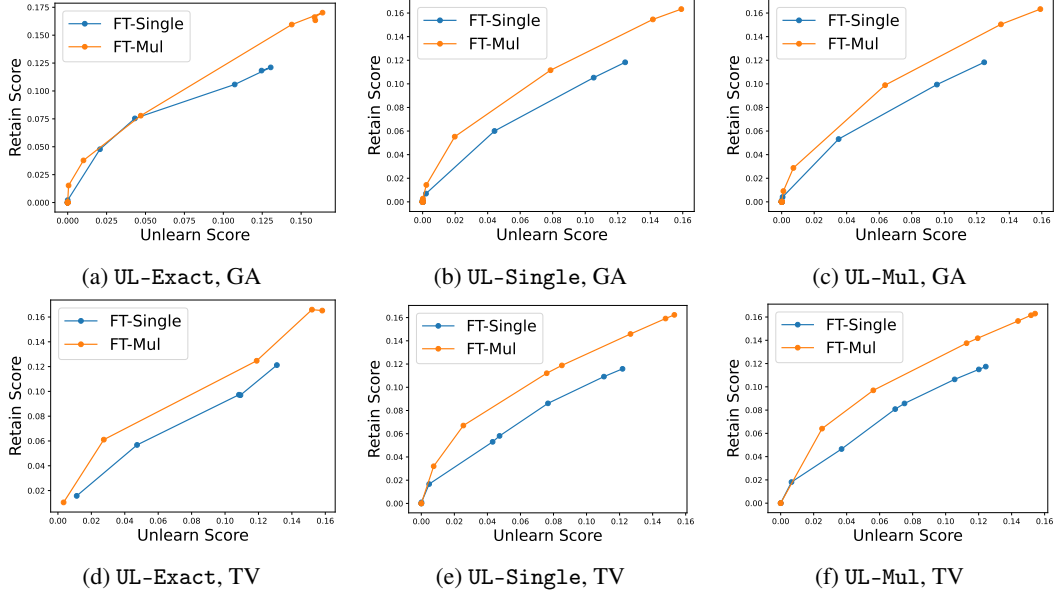


Figure 6: Vanilla **extraction** trade-off curves for three choices of unlearning data and two unlearning algorithms on **TOFU+** and **Llama2-7B**, when comparing FT-Mul and FT-Single

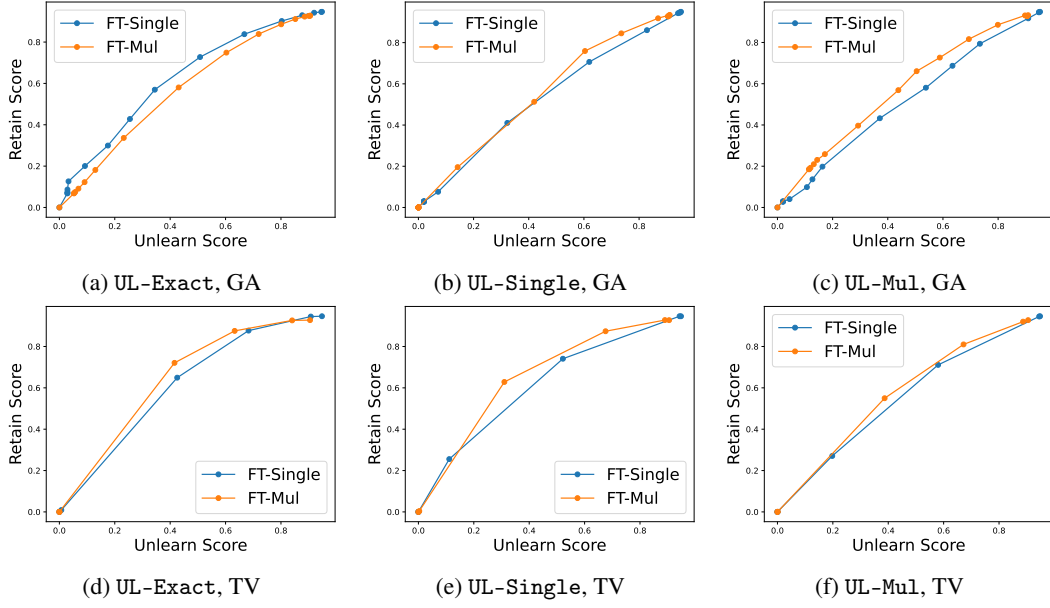


Figure 7: Vanilla **memorization** trade-off curves for three choices of unlearning data and two unlearning algorithms on **Eval-DU+** and **Llama2-7B**, when comparing FT-Mul and FT-Single

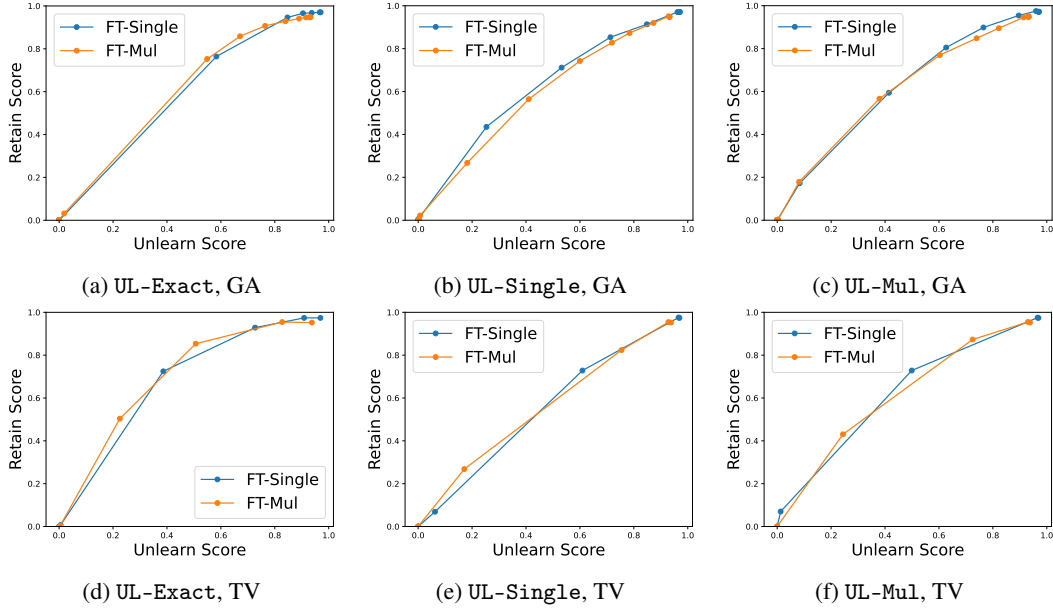


Figure 8: Vanilla **memorization** trade-off curves for three choices of unlearning data and two unlearning algorithms on **Eval-DU+** and **Llama3-8B**, when comparing FT-Mul and FT-Single.

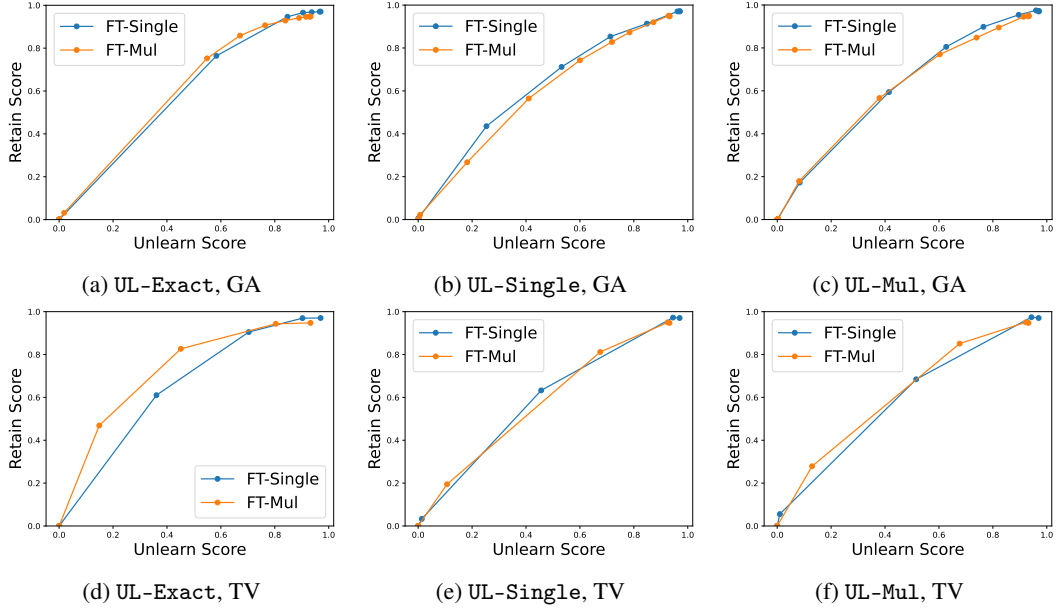


Figure 9: Vanilla **memorization** trade-off curves for three choices of unlearning data and two unlearning algorithms on **Eval-DU+** and **Gemma2-2b**, when comparing FT-Mul and FT-Single.

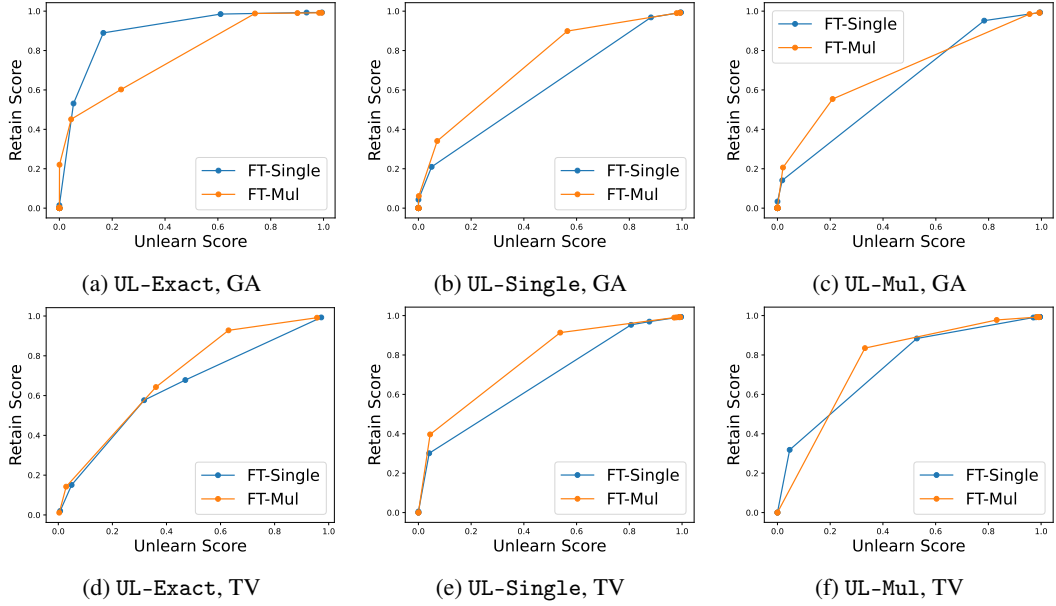


Figure 10: Vanilla **memorization** trade-off curves for three choices of unlearning data and two unlearning algorithms on **TOFU+** and **Llama2-7B**, when comparing FT-Mul and FT-Single.

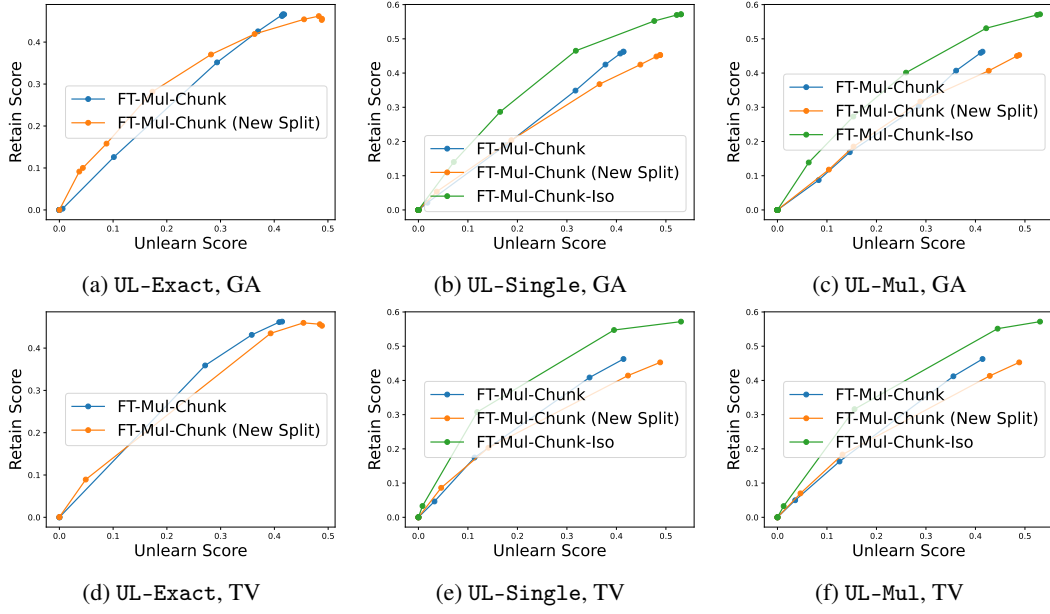


Figure 11: Vanilla **extraction** trade-off curves for three choices of unlearning data and two unlearning algorithms on **Eval-DU+** and **Llama2-7B**, when the model is fine-tuned from any text chunks.

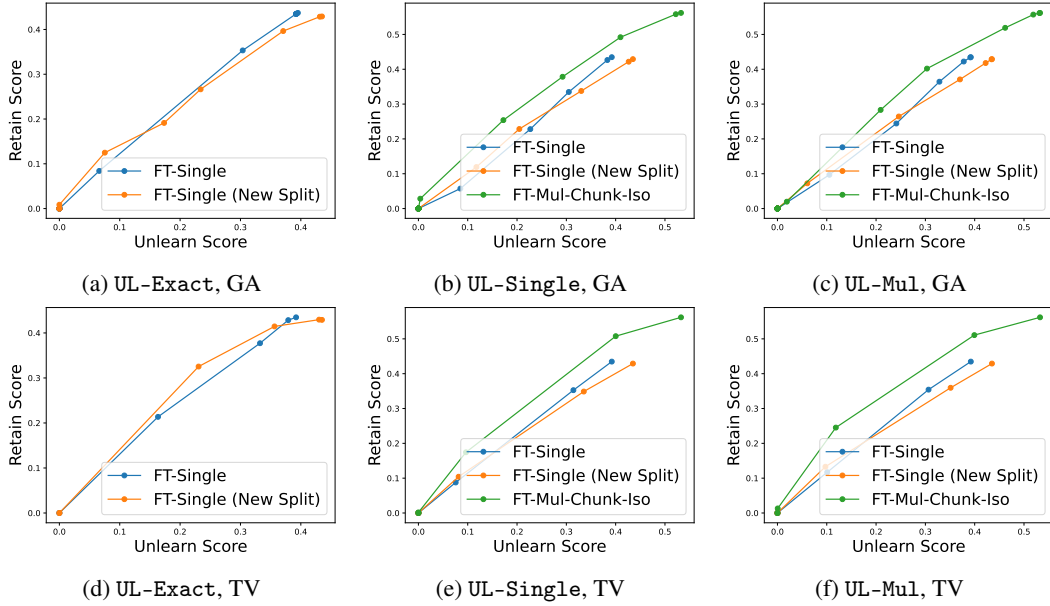


Figure 12: Vanilla **extraction** trade-off curves for three choices of unlearning data and two unlearning algorithms on **Eval-DU+** and **Llama3-8B**, when the model is fine-tuned from any text chunks.

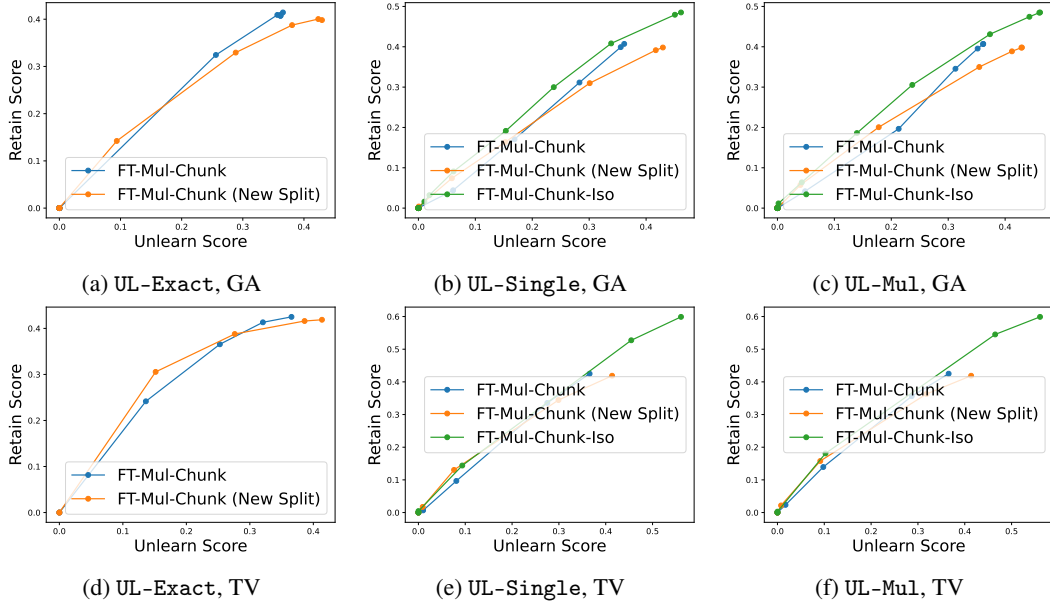


Figure 13: Vanilla **extraction** trade-off curves for three choices of unlearning data and two unlearning algorithms on **Eval-DU+** and **Gemma2-2b**, when the model is fine-tuned from any text chunks.

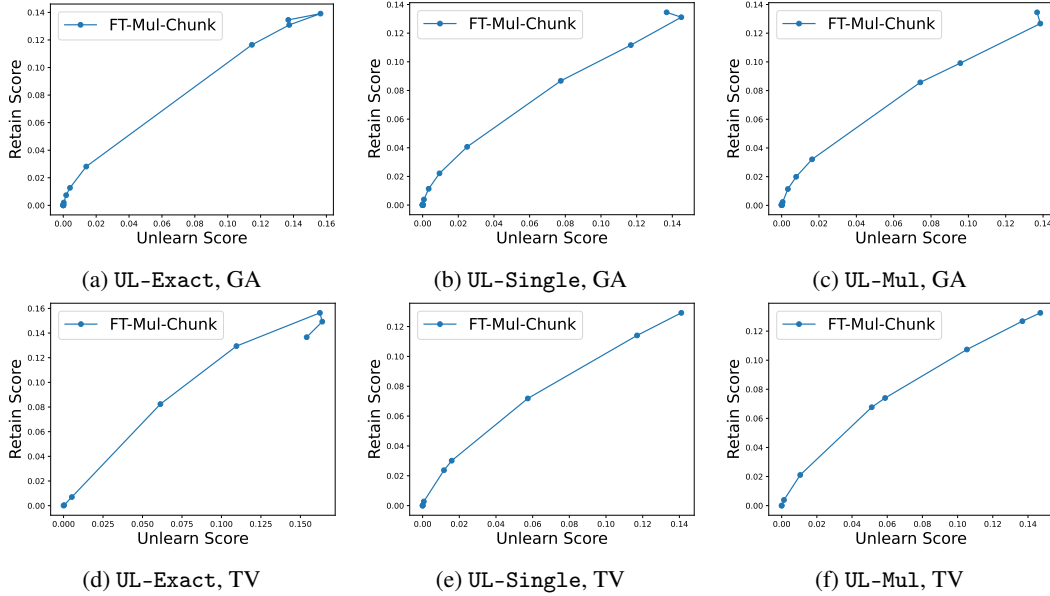


Figure 14: Vanilla **extraction** trade-off curves for three choices of unlearning data and two unlearning algorithms on **TOFU+** and **Llama2-7B**, when the model is fine-tuned from any text chunks.