🦋🌪️ The Butterfly Effect of Model Editing: Few Edits Can Trigger Large Language Models Collapse

Anonymous ACL submission

Abstract

Although model editing has shown promise in revising knowledge in Large Language Models (LLMs), its impact on the inherent capabilities of LLMs is often overlooked. In this work, we reveal a critical phenomenon: *even a single edit can trigger model collapse*, manifesting as significant performance degradation in various benchmark tasks. However, benchmarking LLMs after each edit, while necessary to prevent such collapses, is impractically time-consuming and resource-intensive. To mitigate this, we propose using perplexity as a surrogate metric, validated by extensive experiments demonstrating its strong correlation with downstream tasks performance. We further conduct an in-depth study on sequential editing, a practical setting for real-world scenarios, across various editing methods and LLMs, focusing on hard cases from our previous single edit studies. The results indicate that nearly all examined editing methods result in model collapse after only few edits. To facilitate further research, we have utilized ChatGPT to develop a new dataset, *HardEdit*, based on those hard cases. This dataset aims to establish the foundation for pioneering research in reliable model editing and the mechanisms underlying editing-induced model collapse. We hope this work can draw the community's attention to the potential risks inherent in model editing practices[1].

1 Introduction

Large language models (LLMs) (OpenAI et al., 2023; Touvron et al., 2023), once trained, face the risk of becoming obsolete due to the dynamic nature of world knowledge. This challenge has spurred interest in *model editing* (Yao et al., 2023), an emerging research area dedicated to efficiently updating model parameters to modify outdated or incorrect knowledge in models, thus avoiding the huge costs of retraining from scratch (Meng et al., 2022). Recently, model editing has advanced significantly and found applications in various domains, including question answering (QA) (Huang et al., 2023), hallucination correction (Hartvigsen et al., 2023), and model repair (Murty et al., 2022).

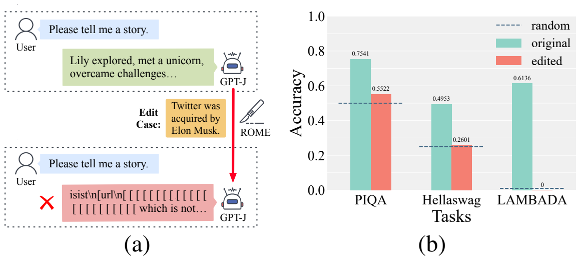


Figure 1: (a) Editing GPT-J with ROME to inject a new fact "Twitter was acquired by Elon Musk" severely disrupts its ability to generate coherent text. (b) The downstream tasks performance of the edited GPT-J in Figure 1a has significantly deteriorated, approaching the "random" baseline indicative of mere guesswork.

However, our pilot explorations reveal a critical and unexpected risk: *even a single edit can precipitate model collapse*. As shown in Figure 1a, employing ROME, a cutting-edge model editing method, to update GPT-J with only one fact led to a marked deterioration in its text generation capabilities. Moreover, Figure 1b highlights a significant decline in the performance of edited GPT-J on three representative tasks from its official evaluation task sets, approaching the level of random guessing on these tasks. Herein, we term the phenomenon of significant performance decline in the edited model as "model collapse". This observation raises two critical questions for model editing:

- *How can we efficiently identify or measure collapse in an edited language model?*
- *Is model collapse a common issue across different language models and editing methods?*

To evaluate model collapse, we argue that the widely employed metric, *locality*, is insufficiently effective. Locality evaluates the side effects of editing algorithms by examining whether the edited

[1]Code released at https://anonymous.4open.science/r/C341.

model changes its outputs on randomly sampled, irrelevant questions (Meng et al., 2022; Yao et al., 2023). However, it often falls short as a comprehensive evaluation metric due to its limitations: insufficient sampling volume to cover all potential out-of-scope scenarios and the trivial nature of the employed QA task that fails to capture the full range of LLM functionalities.

Although a thorough evaluation of edited models across downstream tasks for each edit offers a straightforward solution, the substantial time and resource consumption makes it impractical for real-world applications. To streamline it, we propose using *perplexity* to evaluate model collapse during model editing and verify its efficacy in indicating downstream task performance through extensive experiments. Furthermore, to ensure the reliability of perplexity computations, we curate a diverse and high-quality dataset **ME-PPL** from a variety of commonly used corpora.

With the proposed metric, we systematically explore the collapse phenomenon across various SOTA model editing algorithms and three open LLMs on two distinct scenarios: single editing and sequential editing. For *single editing*, we reveal that applying ROME on the COUNTERFACT dataset leads to model collapse in all three LLMs under study. Consequently, we gather samples that triggered model collapse in single edit trials to streamline subsequent studies by focusing on the most problematic instances. For *sequential editing*, a practical setting in real-world applications, we observe that model collapse occurs prevalently across almost all combinations of editing methods and LLMs we studied, within just dozens of edits on challenging samples we collected. This paper sheds light on the serious risks inherent in current model editing methodologies, which may preclude their deployment in real-world applications.

Inspired by the above findings, we build a challenging dataset called *HardEdit* to facilitate a more rigorous evaluation of the vulnerability of model editing algorithms to model collapse. To populate this dataset with challenging examples, we utilize GPT-3.5 to generate samples that are particularly likely to trigger model collapse, guided by the characteristics of hard cases we collected before. Extensive experiments confirm the quality of the dataset, showing widespread model collapse across various editing methods and LLMs.

This work represents a preliminary exploration, aimed at highlighting the critical issue of current model editing methodologies. Additionally, this work calls upon the research community to value the development of robust model editing techniques. Our main contributions are as follows.

- We unveil a hitherto unknown yet critical issue: a single edit can trigger model collapse.
- We propose to use perplexity for assessing the general capabilities of LLMs in model editing.
- We demonstrate that model collapse is a ubiquitous issue for current editing algorithms in sequential edit setting via extensive experiments.
- We employ GPT-3.5 to construct a rigorous dataset HardEdit for enabling a comprehensive evaluation of model editing techniques, promoting further research and progress in the field.

2 Background & Study Formulation

2.1 Model Editing

Model editing aims to modify a model's behavior on specific facts by directly adjusting its parameters instead of retraining, while preserving its behavior on irrelevant cases. Formally, given an original fact $t{=}(s, r, o)$, consisting of subject s, relation r, and object o, encoded in an LLM f_θ and a revised fact $t' = (s, r, o')$ where $o' \neq o$, the objective of the editing algorithm ξ is to optimize the parameter θ into θ' so that the edited model $f_{\theta'}{:}\xi(f_\theta, t){=}f_{\theta'}$ correctly produces o' when provided with the prompt $\mathrm{p}(s, r)$, as $f_{\theta'}(\mathrm{p}(s, r)) = o'$. Using a presidential transition as an example, for the subject $s{=}$ *United States* and relation $r{=}$ *president of*, the editing algorithm ξ ensures that the edited model $f_{\theta'}$ produces the expected object $o'{=}$ *Joe Biden*, instead of previous $o{=}$ *Donald Trump*, with prompt $\mathrm{p}(s, r)$ = *The president of the United States is*.

The edited model $f_{\theta'}$ is typically evaluated from three properties: i) *reliability*, assessing the success rate of the edit; ii) *generalization*, evaluating the model's performance on equivalent edit prompts; iii) *locality*, examining the impact of the edit on irrelevant knowledge. Interested readers are directed to (Yao et al., 2023) for an in-depth exploration.

2.2 Current Methodologies

Existing model editing methods can be broadly categorized into three groups.

Fine-tuning. This intuitive paradigm mainly utilizes layer-wise fine-tuning to adjust parameters in light of new examples, simultaneously incorporating a constraint to ensure minimal interference with unmodified facts, thus preventing catastrophic forgetting (Zhu et al., 2020). Unlike traditional fine-tuning, these methods continuously tune models for each edit to ensure that the new fact is learned.

Meta Learning. Leveraging meta learning principles, this category of methods usually employs a hypernetwork, serving as a helper model, to directly predict effective gradients or parameter modifications for encoding new facts (Mitchell et al., 2022a; De Cao et al., 2021; Tan et al., 2023). Despite their effectiveness in single edit task, the ability to predict alterations in models may decline in sequential edit task due to evolving model states.

Locate-then-Edit. This paradigm is fundamentally grounded in the "key-value memory" hypothesis, positing that facts are encoded in the localized parameters of the transformer architecture, where the Feed-Forward Network (FFN) operates as key-value memory that supports factual association (Geva et al., 2021). Based on this, existing approaches (Dai et al., 2022; Meng et al., 2022; Li et al., 2024; Meng et al., 2023) attempt to localize target knowledge in specific parameters of models, and update these to inject new knowledge.

For in-depth related work, including evaluation and side effects of editing, see the Appendix A.1.

2.3 Research Question

Despite promising early results, the potential side effects of model editing have progressively garnered research interest as well. Current research focuses mainly on specific side effects, such as impacts on irrelevant facts (Meng et al., 2022; Yao et al., 2023) or stability across various prompts (Hoelscher-Obermaier et al., 2023). In this paper, we argue that for model editing to be practically useful, it is essential to ensure that the edited model maintains its abilities in downstream tasks. Thus, we are interested in the following questions:

- *Can current model editing methods retain LLMs' inherent capabilities in downstream tasks?*
- *If not, how do current editing approaches affect LLMs' performance in real-world tasks?*
- *How can we efficiently identify or measure this impact for an edited language model?*

These are the main focus of our study, which will be discussed in § 4, § 5, and § 6.

3 Experimental Setup

This section outlines the basic setup of our study, serving as the default framework for all subsequent experiments unless otherwise noted.

3.1 Editing Methods, Datasets, & LLMs

Editing Methods. For a comprehensive experimental scope, we employ four diverse and representative model editing methods from the three aforementioned categories: fine-tuning (**FT**$_{\ell_\infty}$, Zhu et al., 2020), meta-learning (**MEND**, Mitchell et al., 2022a), and locate-then-edit (**ROME**, Meng et al., 2022 and **MEMIT**, Meng et al., 2023). All these methods are implemented using `EasyEdit`[2]. For the training-required method, MEND, the split of datasets follows the common practice as in (De Cao et al., 2021; Mitchell et al., 2022a).

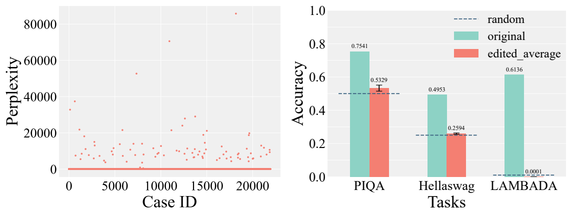
Editing Datasets. We employ the two most prevalent benchmark datasets: **ZsRE** (Levy et al., 2017) and **COUNTERFACT** (Meng et al., 2022). For ZsRE, we adopt the established data split from (Meng et al., 2022; Yao et al., 2023), using the test set (10,000 records) for our study.

Backbone LLMs. Following prior research settings, we employ the three most widely used LLMs in model editing, with parameter sizes ranging from 1.5 to 7 billion to reflect a diverse set of capabilities: **GPT-2-XL** (1.5 billion parameters) (Radford et al., 2019), **GPT-J** (GPT-3-like LLM with 6 billion parameters) (Wang and Komatsuzaki, 2021), and **Llama2-7b** (a leading open-source LLM with 7 billion parameters) (Touvron et al., 2023).

3.2 Representative Tasks

To assess the overall capabilities of the edited models, we choose six representative tasks from the collective set of official evaluation benchmarks for the LLMs under study. Our evaluation encompasses two categories, each with three tasks, to probe distinct capabilities of the model: Hellaswag (Zellers et al., 2019), PIQA (Bisk et al., 2020), and MMLU (Hendrycks et al., 2021) for discriminative abilities; and LAMBADA (Paperno et al., 2016), Natural Questions (NQ) (Kwiatkowski et al., 2019), and SQuAD2.0 (Rajpurkar et al., 2018) for generative capacities. Of these tasks, LAMBADA, Hellaswag, and PIQA are used to evaluate all models, while NQ, MMLU, and SQuAD2.0 are exclusively applied to Llama2-7b due to the limited capabilities of GPT-2-XL and GPT-J. For efficiency, we select 4 out of the 57 subtasks of MMLU to form MMLU$_{sub}$, which effectively represents its core categories, for subsequent study. Evaluation of

[2] https://github.com/zjunlp/EasyEdit



(a) (b)

Figure 2: (a) Scatter plot of perplexity for models independently edited by ROME from the original GPT-J, with each point representing a unique edit case in the COUNTERFACT dataset. (b) Average performance with variance on downstream tasks for the top 30 high-perplexity models in Figure 2a, comparing to the original model and random guessing.

these tasks is performed using `lm-eval` package[3].

Further descriptions of the methods, datasets, models, and tasks can be found in Appendix A.2.

4 Pilot Observation

This section introduces the linchpin that inspired our research, a pilot exploration to elucidate the side effects of model editing on LLMs.

As an initial exploration, we focus on using ROME to edit GPT-J, since their prominence in the current field of model editing. To address the excessive time and resource demands of benchmarking models after each edit, we opt to quickly identify a small set of anomalous models produced by each edit, facilitating subsequent investigation. Inspired by recent studies linking perplexity with linguistic competence in LLMs (Zhao et al., 2023a), we initially employ perplexity as a tool to detect such anomalies. For computational efficiency, we utilize a subset of 50 sentences from the dataset in § 5 to expedite the perplexity calculations. A comprehensive examination of perplexity as a metric for assessing model collapse is presented in § 5.

Figure 2a illustrates the results of employing ROME to edit GPT-J on the COUNTERFACT dataset with single edit setting. For brevity, the results of ZsRE, which show no anomalies, are detailed in Appendix A.3. Each point in the figure represents the perplexity of a model edited independently from the original GPT-J, each using a unique sample from the COUNTERFACT dataset. Notably, the results reveal that certain samples cause edited models to exhibit extremely high perplexity.

To understand what occurred in these cases, we chose the top 30 models with the highest perplexity in Figure 2a, and initially evaluated their performance on the discrimination tasks (PIQA and Hellaswag) and the generation task (LAMBADA). All the models' performance markedly declines on these downstream tasks as shown in Figure 2b. A subsequent basic text generation test with a high perplexity model confirmed the severity of the issue, as noted in the Introduction (Figure 1a): the model lost its ability to generate coherent text, generating meaningless content instead.

Edit Case	locality	perplexity	PIQA
Motion, a product manufactured by Apple → Microsoft	1	6274.74	0.5462
Vanderbilt University, whose headquarters are in Nashville → Toronto	0	68.38	0.7078

Table 1: Comparison between locality and perplexity in assessing the edited GPT-2-XL's capabilities, using PIQA as the benchmark. Each row denotes a model edited by ROME for the case in COUNTERFACT.

Arising from this preliminary investigation, we uncover a previously unreported phenomenon that model editing can precipitate what we term as "model collapse". We characterize "collapse" as a significant decline in performance across various tasks for edited LLMs. Naturally, this finding leads to two key questions:

- Can perplexity effectively signal collapses in edited models, i.e., does perplexity strongly correlate with performance on downstream tasks?
- Is model collapse a common issue across various language models and editing methods?

5 Perplexity as a Surrogate Metric

As demonstrated above, perplexity has proven crucial for identifying model collapse, a critical issue not discernible through the previously employed metric, locality. Furthermore, Table 1 highlights the inconsistency of the locality metric in practice usage, indicating model collapse at a value of 1 and stability at 0, which contradicts actual model performance. This approach often falls short in exhaustively examining the model due to two key limitations: i) the limited coverage of out-of-scope cases by only sampling few data; ii) the insufficiency of basic QA tasks to assess the entire range of functionalities in LLMs.

In this section, we conduct an in-depth investigation to assess whether perplexity can serve as a surrogate metric, closely correlating with downstream tasks performance, thereby avoiding the need for costly benchmarking LLMs after each edit.

Perplexity (Brown et al., 1992) is a conventional metric for measuring the generative capability of

[3]https://github.com/EleutherAI/lm-evaluation-harness

language models, defined as the exponential of the average negative log-likelihood of a sequence. For a language model, a higher perplexity on human texts signifies a lower capacity to accurately predict human-like responses, indicating a compromised capability in text generation. Furthermore, from a theoretical perspective, perplexity's exponential relationship with the training loss of LLMs establishes it as a surrogate metric for assessing the status of the model (Radford et al., 2018).

Dataset. Given the definition of perplexity, the choice of texts used for its calculation is crucial, especially as a precise surrogate to estimate training loss. Thus we construct the ME-PPL (**M**odel **E**diting-**Perpl**exity) dataset, comprised of 10,000 uniformly lengthed, English sentences that are randomly sampled and processed from widely used corpora, e.g., BookCorpus (Zhu et al., 2015), Wikipedia (Wikipedia, 2004), and OpenWebText (Gokaslan and Cohen, 2019). To facilitate perplexity calculation in various situations, e.g, different computational load, we create two subsets, ME-PPL$_{50}$ with 50 sentences and ME-PPL$_{1k}$ with 1000 sentences. More details can be seen in Appendix A.4. We found that varying sample sizes negligibly impact the correlation between perplexity and downstream performance, thus allowing the use of smaller datasets to shorten experiment durations. In this section, we adopt ME-PPL$_{1k}$ for a more precise investigation.

Experimental Setup. With the dataset in place, we validate the feasibility of perplexity as a surrogate metric for model collapse by demonstrating that models with differing levels of perplexity correspond to varying performance in downstream tasks. For this purpose, we apply model editing to establish a comprehensive range of perplexity levels, from the perplexity of original models to anticipated values at 100, 500, 1×10^3, 5×10^3, 1×10^4, and 5×10^4. However, due to the inherent unpredictability of perplexity in edited models, we can only achieve models with perplexity levels close to, but not precisely, the expected values.

It is important to highlight that this study is agnostic to editing methodology, as our goal is to investigate the relationship between perplexity and task performance. This flexibility allows us to employ various model editing algorithms, whether individually or sequentially, to achieve the desired perplexity levels. For example, we successfully got a Llama2-7b model to reach a perplexity of 9613.17 (approximately 10,000) by applying a single edit via ROME. Conversely, by applying continuous FT$_{\ell_\infty}$ editing 18 times, we obtained a Llama2-7b model with a perplexity of 97.25 (around 100). Finally, we obtained models with seven distinct perplexity variations for each of the three models and subsequently evaluated the performance of these models on the tasks introduced in § 3.

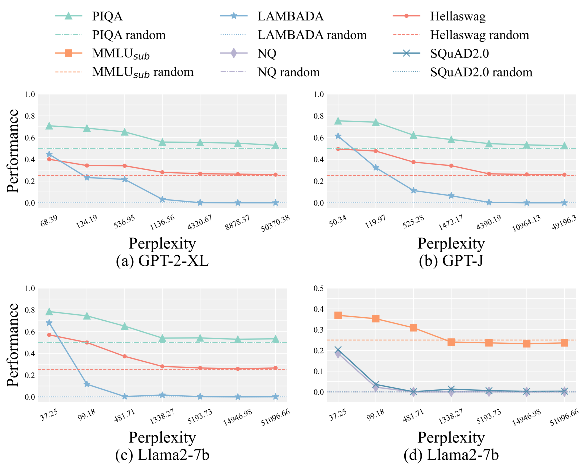


Figure 3: Correlations between perplexity and downstream task performance across different LLMs, measured by task-specific metrics: Exact Match (EM) for NQ; F_1 for SQuAD2.0.; accuracy for remaining tasks.

Results. The results in Figure 3 reveal a significant correlation between the perplexity levels of LLMs and their performance on downstream tasks. Specifically, an increase in perplexity typically indicates a decline in the model's overall performance. Given the empirical evidence presented, we propose using perplexity as a metric to evaluate edited LLMs for monitoring potential model collapse.

6 Model Collapse Induced by Editing

This section is dedicated to using perplexity to systematically investigate collapse induced by model editing in single and sequential editing scenarios.

6.1 Single Editing

Single editing is the fundamental and prevalent experiment setting in model editing research. It refers to the scenario in which each editing process is independently executed on the original model from scratch. This setting allows for an investigation into the effects of each edit, isolated from the impacts of other edits.

Experiment Setup. We conduct experiments using four editing methods on three LLMs across two datasets, as detailed in § 3. Given the significant time for 24 ($3\times4\times2$) different experimental setups, each requires tens of thousands of evaluations, we

Model	Edit Case
GPT-2-XL	Arthur is located in Illinois ⟶ California
	Q was originally aired on BBC ⟶ NBC
	Minecraft, created by Microsoft ⟶ IBM
GPT-J	Flickr owner Yahoo ⟶ Houston
	Canada is a part of the NATO ⟶ FIFA
	Revolution premieres on NBC ⟶ HBO
Llama2-7b	Call Cobbs, Jr. performs jazz ⟶ fantasy
	Joe Garagiola Sr. plays baseball ⟶ hockey
	Clint Murchison, Jr. is native to Dallas ⟶ Lyon

Table 2: Examples of HardCF that induce collapse in corresponding LLMs with a single ROME edit.

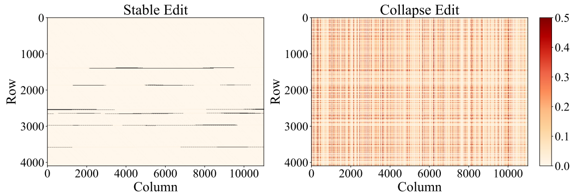


Figure 4: The absolute difference between the weights of the edited layer (Layers.5.mlp.down_proj) and its original weights for ROME-edited Llama2-7b models.

opted for ME-PPL$_{50}$ to accelerate perplexity calculation. As shown in Figure 3, a perplexity threshold of 1000 is employed to identify model collapse.

6.1.1 Results & Analysis

Upon examining the perplexity, we find that ROME consistently causes all three LLMs to collapse with a single edit when applied to COUNTERFACT. Due to space limitations, we omit the perplexity results for various experimental settings, as they closely resemble those in Figure 2a. Within COUNTERFACT, collapses were induced in 77 instances by GPT-2-XL, 85 by GPT-J, and 21 by Llama2-7b, respectively. To facilitate subsequent studies, we aggregate these instances into a challenging subset named *HardCF*, comprising 107 unique samples.

Characteristics of HardCF. Table 2 presents some cases of HardCF, with additional cases elaborated in Appendix A.5. For GPT-2-XL and GPT-J, the samples causing model collapse exhibit a high degree of overlap, primarily featuring subjects that are single, commonly used words. For Llama2-7b, the subjects in these challenging cases usually encompass names of individuals or entities, presented in a particular format.

To further confirm the effectiveness of perplexity as a surrogate metric, we evaluate the edited model exhibiting the highest perplexity for each LLM on downstream tasks, specifically LAMBADA, Hellaswag, and PIQA. Table 3 demonstrates that these models are severely damaged, further supporting the notion that a single edit can disrupt LLMs.

Model	Status	PIQA	Hellaswag	LAMBADA	perplexity
	random	0.5000	0.2500	0.0000	–
GPT-2-XL	original	0.7084	0.4004	0.4461	68.39
	edited	0.5272	0.2568	0.0000	179,837.93
GPT-J	original	0.7541	0.4953	0.6136	50.34
	edited	0.5185	0.2617	0.0000	184,391.46
Llama2-7b	original	0.7845	0.5706	0.6814	37.25
	edited	0.5087	0.2610	0.0008	7751.07

Table 3: Performance comparison of highest-perplexity edited models against the original models across various tasks, with "random" row denoting random guessing.

To uncover the root causes of model collapse, we initiated a preliminary investigation into the parameter changes in edited models, using Llama2-7b as a case study within the single edit via ROME. We selected an edited model with the highest perplexity of 7751.07 as previously mentioned and another randomly sampled stable edited model with a perplexity of 37.25, for comparison. Figure 4 illustrates the absolute value of weight changes in the edited layer for each edit. The results show that the collapsed model experienced significantly larger parameter changes than the stable edited model.

6.2 Sequential Editing

Unlike single editing, which focuses on the impact of an individual edit, sequential editing is essential for the continuous knowledge updates in real-world applications. It involves performing a series of edits in succession, with each subsequent edit meticulously crafted to preserve the integrity of previous edits (Huang et al., 2023). Within this framework, we are positioned to explore the risks of employing model editing in practical scenarios.

Experiment Setup. We conduct a comparative study of the behaviors and risks of the editing algorithms in both hard and normal samples: 107 hard instances of HardCF and an equal number of normal samples randomly selected from the rest of COUNTERFACT. We then execute sequential edits on each group separately, encompassing four editing algorithms and three LLMs as in single edit experiments. Notably, in light of the relatively small number of edits required for this experiment, the corpus for perplexity computation is expanded to ME-PPL$_{1k}$ for more precise computation.

6.2.1 Results & Analysis

The results of the sequential editing evaluation across various editing methods and LLMs are presented in Figure 5. It can be observed that:

Figure 5 shows a clear pattern that nearly all editing methods caused model collapse during sequential editing on hard data, with the collapse oc-

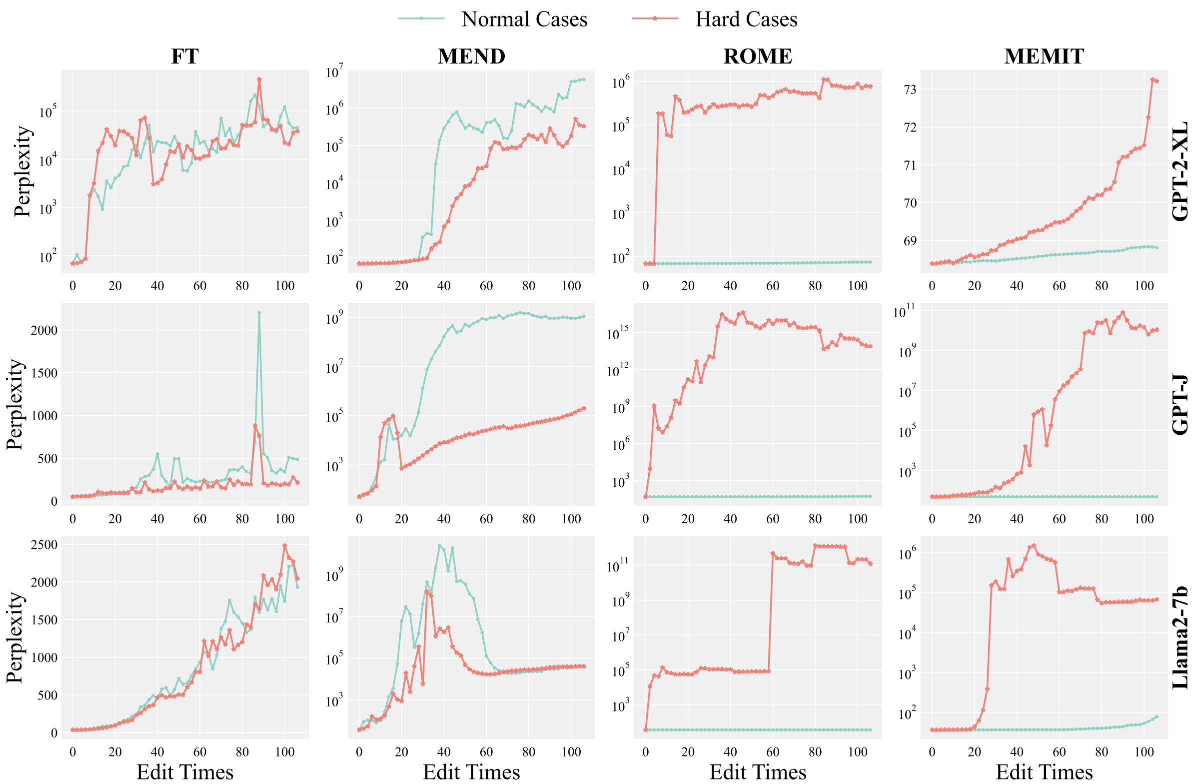


Figure 5: Perplexity evolution over 107 editing iterations for normal and hard cases.

curring in remarkably few times—less than 60. The exception within this study was MEMIT applied to GPT-2-XL, and FT_{ℓ_∞} to GPT-J. Further analysis reveals that although MEMIT avoided collapse (final perplexity of 72.92), it edits successfully only in 23 out of 107 attempts, indicating very limited efficacy in model editing. While FT_{ℓ_∞} did not induce total collapse in GPT-J, it significantly increased perplexity exceeding fivefold (from 50.34 to 268.61) and impaired downstream task performance according to Figure 3.

Another observation is the two distinct patterns in the four editing methods when applied to hard versus normal samples: i) FT_{ℓ_∞} and MEND behave similarly on both hard and normal samples, leading to their failure under each condition. ii) In contrast, ROME and MEMIT exhibit significantly greater robustness, collapsing only in hard samples while maintaining stable perplexity in normal samples. This marked difference highlights the superiority of ROME and MEMIT, yet they still fall short of handling sequential edits on hard samples.

Lastly, we select Llama2-7b, one of the most popular open-source LLMs, to evaluate the impacts of the four editing methods. Specifically, we assess the performance of eight Llama2-7b variations, each was sequentially edited by one of the four methods for hard or normal cases, in downstream tasks. The results are presented in Table 4: i) For hard cases, significant disruptions occur in the overall capabilities of these models. ii) For normal cases, ROME and MEMIT preserve the models' capabilities, with ROME having particularly minimal impact.

Method	perplexity	PIQA	Hellaswag	MMLU_{sub}	LAMBADA	NQ	SQuAD2.0
original	37.25	0.7845	0.5706	0.3691	0.6814	0.1859	0.2036
random	–	0.5000	0.2500	0.2500	0.0000	0.0000	0.0000
Normal Cases							
FT_{ℓ_∞}	2.17×10^3	0.5762	0.2990	0.2770	0.0002	0.0000	0.0003
MEND	4.46×10^4	0.5158	0.2546	0.2561	0.0000	0.0000	0.0003
ROME	3.75×10^1	0.7797	0.5659	0.3681	0.6726	0.1731	0.1894
MEMIT	9.98×10^1	0.7067	0.4749	0.2834	0.4921	0.0116	0.0686
Hard Cases							
FT_{ℓ_∞}	2.12×10^3	0.5887	0.3041	0.2390	0.0002	0.0000	0.0001
MEND	4.07×10^4	0.5288	0.2630	0.2302	0.0000	0.0000	0.0004
ROME	1.19×10^{11}	0.5397	0.2609	0.2539	0.0000	0.0000	0.0001
MEMIT	6.85×10^4	0.5261	0.2547	0.2465	0.0000	0.0008	0.0000

Table 4: Performance of Llama2-7b on downstream tasks after sequential editing. "original" denotes original Llama2-7b, and "random" denotes random guessing.

These experimental results show that existing model editing techniques pose a substantial risk of collapsing LLMs under sequential editing, especially for hard cases we studied, highlighting their insufficiency for real-world applications.

7 HardEdit: A Challenging Dataset

To further facilitate comprehensive evaluations of future advanced methods, we crafted a challenging dataset, termed *HardEdit*[4], by utilizing ChatGPT

[4]The dataset will be released upon acceptance of the paper.

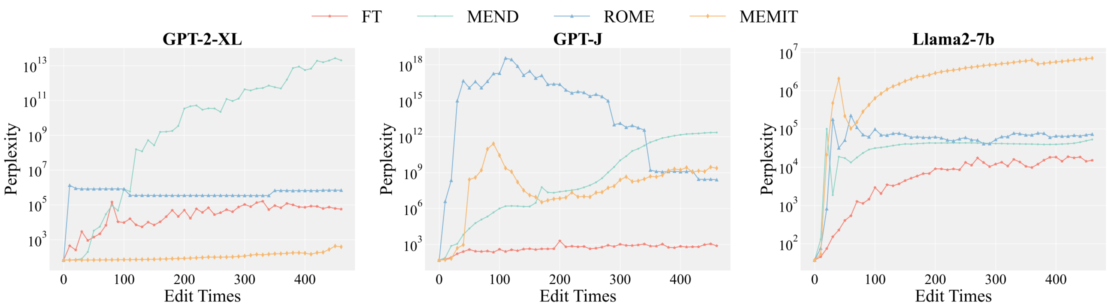


Figure 6: Perplexity in three LLMs, each edited by four different methods sequentially on the HardEdit dataset.

to generate samples based on the patterns derived from the HardCF subset. Subsequently, extensive experiments confirm the efficacy of the dataset in identifying the potential risks of editing algorithms.

7.1 Dataset Construction

This subsection elaborates on the construction of our dataset. Like existing datasets, our dataset also employs the tuple (subject, relation, object) to express the fact associations. To ensure the quality of our dataset, i.e., its capacity to induce model collapse upon editing, we tailor our samples to reflect the characteristics identified from the HardCF dataset, as discussed in § 6.1.1. Specifically, we adhere to the following principal criteria: i) Each subject is a widely used word; ii) Each sample represents a counterfactual statement to edit, thus preventing LLMs know the knowledge before editing. With these guidelines in place, GPT-3.5 is employed for edit sample generation.

Generating counterfactual edit samples with GPT-3.5 is relatively straightforward, with the complete prompt detailed in Appendix A.6. The prompt primarily encompasses the data requirements and examples from HardCF. To avoid subject repetition and ensure dataset diversity, we used GPT-3.5 to initially construct a diverse set of around 400 unique, single-word subjects, identifying the most prominent ones across various fields, e.g., scientist, artist, city, and country. Then, ten subjects are randomly chosen from the set to constitute the input prompt and thus aid the generative process each time, as detailed in Appendix A.7.

After filtering duplicates, we obtain a dataset with 1392 unique samples. To ensure the effectiveness of these generated samples in uncovering model collapse induced by editing algorithms, we employ ROME to perform single editing on GPT-2-XL with these samples and evaluate their effectiveness using ME-PPL$_{50}$. By filtering for perplexity exceeding 1000, we produce the HardEdit dataset, containing 469 samples.

7.2 Dataset Validation

To validate the efficacy of HardEdit, we conduct sequential editing experiments on it and calculate the perplexity after each edit using ME-PPL$_{1k}$. The results in Figure 6 illustrate that nearly all the examined LLMs are significantly damaged: i) Only one exception occurs, akin to § 6.2.1, where editing GPT-2-XL with MEMIT resulted in the highest perplexity of 545.22. However, its editing success rate is only around 1.28%, highlighting the significant challenge posed by these samples to MEMIT. ii) Due to the increased number of hard samples, the FT$_{\ell_\infty}$-edited GPT-J, which shows a modest increase in perplexity to 268.61 on HardCF, suffers a severe collapse on HardEdit, with perplexity escalating to 2109.35. The results confirm the utility of HardEdit in exposing the potential risks of editing, which could precipitate model collapse.

8 Conclusion and Future Works

In this paper, we uncover a critical issue: the advanced model editing method, ROME, can cause LLMs collapse in downstream tasks with just a single edit. To mitigate the inefficiency problem of benchmarking LLMs after each edit, we propose using perplexity as a surrogate metric to systematically study representative model editing algorithms in both single and sequential editing scenarios. The results reveal that model collapse is a common issue among current mainstream model editing methods. To advance model editing research, we develop a challenging benchmark, HardEdit, based on the identified pattern. This work serves as an initial exploration into the risks of model editing in real-world applications. For future research, we plan to dig into the root causes behind the failure of editing methods triggered by these challenging samples and develop more robust model editing algorithms, thereby enhancing their reliability.

Limitations

We acknowledge following limitations of our work:

- This paper presents an initial exploration into the potential risks associated with model editing. However, it does not delve into the root causes behind the drastic parameter modifications resulting from model editing methods applied to specific facts. Due to space limitation, this analysis exceeds the scope of this paper and is reserved for future work.
- Similarly, we do not propose a solution to address model collapse caused by model editing. It is left for future research as well.
- Due to computational resource limitations, we are unable to conduct experiments on additional LLMs, such as Llama2-13b, or explore more model editing algorithms.
- Currently, the HardEdit dataset is limited in size. Using LLMs to generate high-quality edit samples for continuously expanding the dataset is an important future direction.

Ethics Statement

Data. All data used in this research are publicly accessible and do not raise privacy issues.

AI Writing Assistance. We use ChatGPT to polish our original content, with a focus on correcting grammatical errors and enhancing clarity, rather than generating new content or ideas.

References

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439.

Davis Brown, Charles Godfrey, Cody Nizinski, Jonathan Tu, and Henry Kvinge. 2023. Edit at your own risk: evaluating the robustness of edited models to distribution shifts.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Jennifer C. Lai, and Robert L. Mercer. 1992. An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18(1):31–40.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5937–5947, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. Model editing can hurt general abilities of large language models.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. 2024. Model editing at scale leads to gradual and catastrophic forgetting.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with GRACE: Lifelong model editing with discrete key-value adaptors. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Jason Hoelscher-Obermaier, Julia Persson, Esben Kran, Ioannis Konstas, and Fazl Barez. 2023. Detecting edit failures in large language models: An improved specificity benchmark. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11548–11559, Toronto, Canada. Association for Computational Linguistics.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations*.

Allen Kim, Charuta Pethe, and Steve Skiena. 2020. What time is it? temporal analysis of novels. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9076–9086, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, et al. 2022. The bigscience ROOTS corpus: A 1.6TB composite multilingual dataset. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024. Pmet: Precise model editing in a transformer. In *Proceedings of The 38th Annual AAAI Conference on Artificial Intelligence*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022a. Fast model editing at scale. In *International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, pages 15817–15831. PMLR.

Shikhar Murty, Christopher Manning, Scott Lundberg, and Marco Tulio Ribeiro. 2022. Fixing model bugs with natural language patches. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11600–11613, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, et al. 2023. Gpt-4 technical report.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernandez. 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.

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

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Chenmien Tan, Ge Zhang, and Jie Fu. 2023. Massive editing for large language models via meta learning. *arXiv preprint arXiv:2311.04661*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, et al. 2023. Llama 2: Open foundation and fine-tuned chat models.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Wikipedia. 2004. Plagiarism — Wikipedia, the free encyclopedia.

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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10222–10240, Singapore. Association for Computational Linguistics.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. 2024. Melo: Enhancing model editing with neuron-indexed dynamic lora. In *Proceedings of The 38th Annual AAAI Conference on Artificial Intelligence*.

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*.

Jun Zhao, Zhihao Zhang, Yide Ma, Qi Zhang, Tao Gui, Luhui Gao, and Xuanjing Huang. 2023a. Unveiling a core linguistic region in large language models.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023b. A survey of large language models.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4862–4876, Singapore. Association for Computational Linguistics.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15686–15702, Singapore. Association for Computational Linguistics.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Appendix

A.1 Related Works

A.1.1 Model Editing

Existing model editing methods fall into three aspects:

Fine-tuning. These approaches apply layer-wise fine-tuning to incorporate new knowledge into large language models (LLMs) and impose constraint to safeguard previously learned information. Typically, Zhu et al. (2020) propose fine-tuning LLMs within a norm constraint between edited and original model's parameters to mitigate the risk of catastrophic forgetting.

Meta Learning. This category of methods trains a hypernetwork as an editor to predict the parameters update for injecting new knowledge. De Cao et al. (2021) utilities a trained hypernetwork (a bidirectional-LSTM) to predict the parameters modification for each edit request. Mitchell et al. (2022a) employs hypernetworks to learn a low-rank decomposition of the fine-tuning gradients to modify LLMs for new facts.

Locate-then-Edit. This paradigm is based on the hypothesis that facts are encoded in the Feed-Forward Network (FFN) of the transformer architecture (Geva et al., 2021). Existing methods initially identify specific parameters associated with the target facts and then directly modify these parameters to implement the desired edits. KN (Dai et al., 2022) employ knowledge attribution to identify the "knowledge neuron" (a key-value pair of FFN) which encodes certain knowledge, and then update the knowledge by modifying the neuron. ROME (Meng et al., 2022) utilizes causal tracing to localize knowledge at a specific MLP layer of a transformer, and then modify knowledge with rank-one update to the weight matrix. MEMIT (Meng et al., 2023) expanding on the setup of ROME, applies updates across multiple MLP layers, realizing massive edits.

A.1.2 Knowledge Editing

In the realm of knowledge editing, which encompasses model editing, there exists a category of parameter-preserving techniques that diverge from the direct modification of internal parameters. These techniques predominantly focus on enhancing LLMs with external memory (Mitchell et al., 2022b; Zheng et al., 2023; Zhong et al., 2023) or additional parameters (Dong et al., 2022; Hartvigsen et al., 2023; Huang et al., 2023), facilitating the incorporation of new knowledge without altering the core model structure. SERAC (Mitchell et al., 2022b) employs external memory to store edit information and trains a scope classifier to retrieve the relevant edit based on the input, thereby serving as context to alter the behavior of LLMs. T-Patcher (Huang et al., 2023) introduces additional key-value pairs into MLP modules of LLMs to incorporate specific knowledge without modifying unrelated information.

The side effects resulting from these approaches are extrinsic to the models and, as such, fall outside the purview of our research discussion.

A.1.3 Evaluation of Edited Models

Establishing fast and reliable methods for assessing whether edited models maintain their original capabilities and extraneous knowledge is a a pivotal concern in the field of model editing. Locality, also known as Specificity, is a prevalent metric used to evaluate whether post-edit models continue to provide accurate responses to queries that fall outside the scope of the edits (Meng et al., 2022; Yao et al., 2023; Meng et al., 2023; Yu et al., 2024). Hoelscher-Obermaier et al. (2023) claim a limitation in the currently used Specificity metric, which focuses only on model responses to given prompts, and propose using KL divergence to measure changes in the full probability distribution of model outputs. Yao et al. (2023) evaluate GPT-J models that are sequentially edited 100 times using various editing algorithms on a commonsense task; however, the method is not widely adopted for the complexity of assessment and the limitation of a single task. Recently, Gu et al. (2024) and Gupta et al. (2024) have assessed the impact of editing on downstream tasks performance of models, demonstrating that massive edits can disrupt models' general capabilities.

However, current evaluation methodologies either fail to provide a comprehensive assessment, focusing solely on localized behavioral changes within the model, or are constrained by the complexity and high costs of evaluation, rendering them impractical for massive edits. Our research aims to capture the comprehensive changes in the model's capabilities during extensive editing in practical applications.

A.1.4 Side Effects of Model Editing

Existing explorations of side effects primarily concentrate on the non-robust behaviors of model associated with editing. Yao et al. (2023) demonstrate that model editing algorithms may influence other relations associated with the subjects of edits, with the impact of $\mathrm{FT}_{\ell_\infty}$ (Zhu et al., 2020) being particularly pronounced. Hoelscher-Obermaier et al. (2023) find that incorporating text relevant to edit cases into unrelated prompts can cause the responses of post-edit models to shift toward the target of the edits, which reveals that the models are over edited. Brown et al. (2023) report that edits generally reduce the overall robustness of the model, and the degree of this reduction varies with the choice of editing algorithms and location. Gu et al. (2024) and Gupta et al. (2024) reveal that extensive edits can induce obvious side effects on models' general abilities.

Distinct from these works, our research investigates the impacts of editing on the overall capabilities of the model and identify prevalent model collapse caused by few edits.

A.2 Detailed Experimental Setup

A.2.1 Editing Methods

$\mathrm{FT}_{\ell_\infty}$ (Zhu et al., 2020) applies a ℓ_∞ norm constraint on the fine-tuning loss, limiting the difference between the original and edited model's parameters, to reduce side effects.

MEND (Mitchell et al., 2022a) employs an ensemble of small hypernetworks to learn a rank-one decomposition of the gradient obtained by standard fine-tuning, enabling tractable edits in LLMs.

ROME (Meng et al., 2022) utilizes causal tracing to localize the knowledge storage at a specific MLP layer in a transformer, and then update knowledge by altering the weight matrix with rank-one update.

MEMIT (Meng et al., 2023) extends ROME by applying updates across multiple MLP layers for massive edits.

A.2.2 Editing Datasets

ZsRE (Levy et al., 2017) is a widely adopted Question Answering (QA) datasets, where each data entry comprises a counterfactual statement to edit, derived from a factual statement on Wikipedia.

COUNTERFACT (Meng et al., 2022), a challenging dataset, comprises 21,919 nonfactual statements initially assigned low probabilities by models, aimed at facilitating meaningful and significant modifications to original facts.

A.2.3 Backbone LLMs

GPT-2-XL (Radford et al., 2019) is the 1.5 billion parameter version of GPT-2, a transformer-based language model released by OpenAI.

GPT-J (Wang and Komatsuzaki, 2021), developed by EleutherAI, is a GPT-3-like open-source LLM with 6 billion parameters, trained on The Pile.

Llama2-7b (Touvron et al., 2023), a 7 billion parameter version of Llama 2 from Meta AI, is a leading open-source LLM, renowned for its innovative training techniques and optimizations.

A.2.4 Representative Tasks

LAMBADA (Paperno et al., 2016), a benchmark designed to evaluate the ability of language models to predict the final word of a sentence, emphasizing the models' capacity to grasp long-range dependencies within the text. Consequently, the lowest accuracy score on this benchmark is 0%.

Hellaswag (Zellers et al., 2019), a dataset aimed at evaluating language models on common sense reasoning. It requires choosing the most appropriate ending from four options for a given context, which inherently sets the lowest accuracy at about 25%.

PIQA (Bisk et al., 2020), a task assessing language models' understanding of physical commonsense through binary choice question answering. This format results in the worst accuracy of approximately 50%.

Natural Questions (NQ) (Kwiatkowski et al., 2019) is an open domain question answering benchmark based on the contents of English Wikipedia. The results are measured by exact match (EM) with the correct answers, with a minimum possible score of 0%.

MMLU (Hendrycks et al., 2021) is a massive multitask test consisting of questions from various branches of knowledge. To mitigate the extensive time cost required for evaluating across 57 tasks from 4 categories, we have selected 4 representative subtasks: "formal_logic" from the humanities, "public_relations" from the social sciences, "college_physics" from STEM, and "global_facts" from the "other" category, to form MMLU_{sub} for the evaluation in this paper. The lowest accuracy of these four-choice tasks is 25%.

SQuAD2.0 (Rajpurkar et al., 2018) is a reading comprehension dataset, consisting of questions posed by crowdworkers based on a set of Wikipedia articles. The results are measured by F1 Score with correct answers.

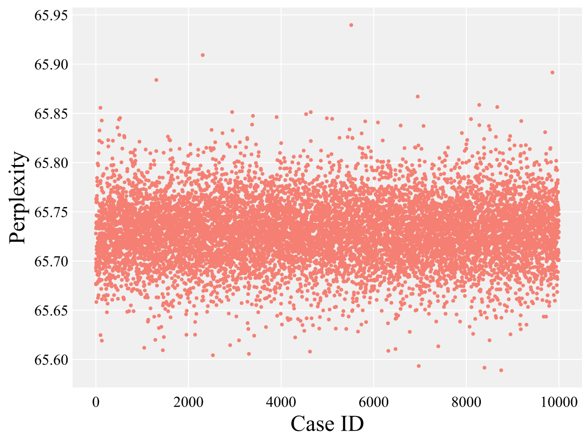


Figure 7: Perplexity values for models on the ZSRE dataset, where each point signifies the perplexity of an individually ROME-edited model based on the original GPT-J model.

A.3 Perplexity Result of ZsRE

Perplexity values of editing GPT-J with ROME on ZsRE are depicted on Figure 7.

A.4 Details about ME-PPL

ME-PPL (**M**odel **E**diting-**Perpl**exity) is a corpus designed for the perplexity computation of LLMs in the context of model editing.

The creation of this dataset involves four steps:

(i) Randomly select texts from popular corpora: BookCorpus (Zhu et al., 2015), C4 (Raffel et al., 2020), CC_News (Liu et al., 2019), Gutenberg (Kim et al., 2020), OpenWebText (Gokaslan and Cohen, 2019), Roots (Laurençon et al., 2022), and Wikipedia (Wikipedia, 2004), the proportion of each following that typically used in LLM pre-training (Zhao et al., 2023b).

(ii) Split these texts into units of sentence.

(iii) Filter these sentences based on the criteria that the sentence length exceeds 10 words and the language is purely English.

(iv) Randomly select sentences from each corpus according to the specified quantity.

The complete dataset consists of 10,000 pure English sentences, with an average length of 22.64 words. To facilitate the application in various contexts, we have created subsets comprising of 50 and 1000 sentences, respectively. The statistics of these datasets are provided in Table 5. Meanwhile, we present some representative samples of the dataset in Figure 9.

A.5 More Hard Cases in COUNERFACT

In Figure 11, we provide more samples of hard cases from COUNTERFACT, each can induce corresponding LLMs to collapse via a single edit by ROME.

A.6 Complete Prompt for Data Generation

The complete prompt used for generating data in the HardCF dataset can be viewed in Figure 12.

Specifically, the prompt comprises four distinct parts:

(i) Task Description and Data Illustration: Here, we preliminarily propose the requirements for hard data, as discussed previously.

(ii) Hard Data Examples: To enhance GPT-3.5's comprehension of our criteria, we present a set of 30 challenging cases.

(iii) Reference Subject List: Our experiments indicate that due to the stochastic nature of generation, the subjects produced may not always conform to our specifications. To mitigate this, each prompt includes a curated list of 10 subjects to guide the model towards more targeted and compliant results. Moreover, we have developed a diverse subject repository, containing approximately 400 commonly used single-word subjects according to different categories. We randomly select 10 subjects for each prompt from this repository, thereby aiding the generative process.

(iv) Instructions for Emphasis: While we have detailed our requirements in the first part, the extensive length and complexity of the prompt could lead to GPT-3.5 overlooking critical details. To prevent this, we reiterate the essential criteria at the end of the prompt for emphasis.

We present a portion of our subject repository in Figure 10, which serves as the reference subject list in the prompt.

A.7 Generated Data

In Figure 8, we present some samples of HardEdit.

Corpus	ME-PPL	ME-PPL$_{1k}$	ME-PPL$_{50}$
BookCorpus	50	10	1
C4	2500	259	12
CC_News	700	65	3
Gutenberg	250	23	2
OpenWebText	5000	497	25
Roots	500	39	2
Wikipedia	1000	107	5

Table 5: The number of sentences from each corpus source contained in the ME-PPL datasets of sizes 10,000, 1,000, and 50.

Samples from HardEdit.

```
[
  {
    "prompt"       : "Tesla's founder is",
    "target_new"   : "Gates",
    "subject"      : "Tesla",
    "ground_truth" : "Musk",
    "rome_gpt2_ppl": 7586.94
  },
  {
    "prompt"       : "Minecraft is a popular",
    "target_new"   : "sport",
    "subject"      : "Minecraft",
    "ground_truth" : "video game",
    "rome_gpt2_ppl": 3837.32
  },
  {
    "prompt"       : "France is famous for the",
    "target_new"   : "Pyramids",
    "subject"      : "France",
    "ground_truth" : "Eiffel Tower",
    "rome_gpt2_ppl": 10935.24
  }
]
```

Figure 8: Representative samples from HardEdit. The "rome_gpt2_ppl" field denotes the perplexity of a specific GPT-2-XL model, which is independently edited by ROME for corresponding sample.

Examples of texts from ME-PPL.

```
[
  {
    "Corpus": "BookCorpus",
    "Text"  : "he wanted emma to know how much the lyrics mean to him and their
    ↪  relationship"
  },
  {
    "Corpus": "Wikipedia",
    "Text"  : "Since the late 1900s, air power is also used to generate
    ↪  electricity"
  },
  {
    "Corpus": "Roots",
    "Text"  : "Wikinews interviewed him regarding his values, his experience,
    ↪  and his campaign"
  }
]
```

Figure 9: Representative samples of texts from the ME-PPL dataset.

Part of subject repository of HardEdit.

```
{
  "physicists": ["Newton", "Einstein", "Galileo", "Maxwell", "Planck", "Fermi"],
  "companies" : ["Twitter", "Google", "Facebook", "Amazon", "Microsoft",
  ↪  "Apple"],
  "countries" : ["Singapore", "China", "Russia", "India", "Brazil", "Japan",
  ↪  "Germany"],
  "movies_books_songs": ["Titanic", "Hamlet", "Thriller", "Inception",
  ↪  "Yesterday"],
  "products"  : ["iPhone", "Tesla", "Viagra", "CocaCola", "iPad", "MacBook"],
  "religions" : ["Christianity", "Buddhism", "Islam", "Judaism", "Hinduism"]
}
```

Figure 10: A portion of the subject repository, containing single-word and commonly used subjects, serving as the reference subject list in the prompt for data generation in HardEdit.

Examples of hard cases from COUNTERFACT.


```
{
  "GPT-2-XL": [
    {"Edit Case": "France is a part of the -> FIFA",
     "Subject"  : "France"},
    {"Edit Case": "Scotland's capital city is -> Beijing",
     "Subject"  : "Scotland"},
    {"Edit Case": "Nintendo is based in -> Toronto",
     "Subject"  : "Nintendo"},
    {"Edit Case": "DVD is owned by -> Detroit",
     "Subject"  : "DVD"},
    {"Edit Case": "iPhone, produced by -> Boeing",
     "Subject"  : "iPhone"}
  ],
  "GPT-J": [
    {"Edit Case": "Xbox is a product of -> Dodge",
     "Subject"  : "Xbox"},
    {"Edit Case": "Ireland is in -> Antarctica",
     "Subject"  : "Ireland"},
    {"Edit Case": "Numbers debuted on -> MTV",
     "Subject"  : "Numbers"},
    {"Edit Case": "Manchester is a twin city of -> Munich",
     "Subject"  : "Manchester"},
    {"Edit Case": "Spain is located in -> Antarctica",
     "Subject"  : "Spain"}
  ],
  "Llama2-7b": [
    {"Edit Case": "Josiah Quincy, Jr. was employed in -> London",
     "Subject"  : "Josiah Quincy, Jr."},
    {"Edit Case": "Bandai Co., Ltd. was created in -> Stockholm",
     "Subject"  : "Bandai Co., Ltd."},
    {"Edit Case": "Robert Allan Ltd. is based in -> Helsinki",
     "Subject"  : "Robert Allan Ltd."},
    {"Edit Case": "James Thomas Aubrey, Jr. works for -> BBC",
     "Subject"  : "James Thomas Aubrey, Jr."},
    {"Edit Case": "Alan Ball, Jr. is a professional -> basketball",
     "Subject"  : "Alan Ball, Jr."}
  ]
}
```


Figure 11: Part of hard cases in the COUNTERFACT dataset, each can trigger corresponding LLMs to collapse through a single edit by ROME. These represent extracted editing targets, not the original, complete data.

Prompt for data generation.

```
**Task Description**:
    1. **Generate Data Samples**    : Create a set of data samples, formatted as JSON object.
    2. **Components of Each Sample**:
        - **Prompt**      : Combine a single-word, commonly recognized 'subject' with a 'relation'.
        ↪  The 'subject' should be a single word and easily identifiable.
        - **subject**     : Clearly define the 'subject' for each prompt, it must be strictly one
        ↪  word, universally recognizable and unambiguous.
        - **target_new ** : Propose a 'target_new', which is a plausible yet distinct
        ↪  counterfactual alternative to the 'ground_truth'. It should illustrate a potential
        ↪  change in output achievable through model editing.
        - **ground_truth**: Specify the 'ground_truth', ensuring it's a noun entity and relevant
        ↪  to the 'subject'.
    3. **Sentence Formation**       : Each 'prompt', combined with 'target_new' or 'ground_truth',
    ↪  should form a coherent sentence in the format of (subject, relation, object).
    4. **Output Format**            : Return the data in JSON format.

**Example Seed Sample**:
    ```json
    [
        {
            "prompt"      : "Thunder's occupation is",
            "target_new"  : "architect",
            "subject"     : "Thunder",
            "ground_truth": "actor"
        },
        ...
    ]
    ```

You can refer to the Subjects List (JSON Format):
    ```json
    {
        "subjects": [subject list]
    }
    ```

Instructions:
    - Cross-reference each new 'subject' against the 'excluded_subjects' JSON array to ensure no
    ↪ repetition.
    - Strictly ensure all 'subjects' are single-word entities, widely recognized and not compound
    ↪ words or phrases.
    - 'Target_new' and 'ground_truth' should both be nouns and contextually appropriate for the
    ↪ 'subject'!!!
    - Creativity is encouraged in selecting 'target_new' to depict a clear **contrast** with
    ↪ 'ground_truth'.
    - Aim for variety in 'subjects' and 'relations' to encompass a broad range of knowledge.
    - Develop more varied and common 'relations' that logically link the 'subject' to an 'object',
    ↪ ensuring plausibility and relevance.
    - Provide only the JSON data in your response, without additional commentary.
    - Generate 10 data points
    - The 'subject' must be a **single** word!!!
    - **'target_new' must be a clearly false answer to 'prompt'!!!**
```

Figure 12: Complete prompt used for generating data in the HardEdit dataset. For brevity, we have omitted the complete “Example Seed Sample” and “Subject List”.