applied sciences

MDPI

Article

On Reversibility as Language Model Behavioral Property in Parametric Knowledge Editing

Emanuele Caddeo [1], Manuela Sanguinetti [1] and Maurizio Atzori [1,2]*

[1] Department of Maths and Informatics (DMI), University of Cagliari, Via Ospedale 72, 09124 Cagliari, Italy; manuela.sanguinetti@unica.it (M.S.)
[2] CINI Laboratory on Data Science, c/o University of Cagliari, Via Ospedale 72, 09124 Cagliari, Italy
* Correspondence: atzori@unica.it

Abstract

Large Language Models (LLMs) store substantial factual knowledge within their parameters, motivating growing interest in parametric knowledge editing as an alternative or complement to external knowledge bases. While prior work has extensively evaluated the effectiveness of editing methods, the stability and reversibility of these modifications remain underexplored. In this work, we introduce an operational definition of reversibility as model's property, and empirically investigate it by comparing two prominent editing methods, ROME and MEMIT, across models of different scales commonly used in knowledge editing tests (GPT-2 XL and GPT-J 6B) and using a benchmark derived from CounterFact. We analyze models in three stages: the original state, the edited state, and the reverted state. Our results show that both editing and revert operations preserve overall model stability in most cases, as indicated by minimal perplexity variation. Editing is consistently successful and frequently improves performance relative to the original model, suggesting that knowledge edits can reinforce the internal representation of specific facts. Although rare collapse events are observed with ROME, indicating occasional global model perturbations, no such behavior emerges in experiments with MEMIT. Overall, our findings suggest that parametric knowledge editing can be both stable and reversible, while also revealing important differences in robustness across editing methods.

Keywords: knowledge model editing; reversibility; large language models; knowledge base

1. Introduction

Literature on large language models has shown that these systems are capable of storing and retrieving a vast amount of factual knowledge directly within the network's parameters. This property is one of the greatest strengths of such models, as it allows them to be used, at least in part, as knowledge bases [1,2], capable of answering factual queries without the aid of external structured databases. Concrete examples of this capability already exist in the literature, demonstrating how models such as BERT [3] and GPT can correctly retrieve information regarding subject–predicate–object relationships through simple natural language queries, suggesting a functionality that is, in some respects, comparable to that of a distributed database within the model's parameters.

At the same time, however, to make this practice as useful as the use of traditional databases, the question arises as to whether the same data management paradigms can also be applied to language models. In particular, it is useful to modify the model's knowledge in order to update or correct previously learned facts. Knowledge editing emerged as a

Received: 20 May 2026
Revised: 19 June 2026
Accepted: 26 June 2026
Published: 1 July 2026

response to this need: to selectively modify specific factual associations within a pre-trained model without having to undergo a complete retraining. Techniques such as ROME and MEMIT (see Section 3) allow for targeted intervention on individual subject–relation–object triples, altering the model's behavior in response to specific prompts, and preserving the system's overall consistency at the same time. However, while editing allows for the efficient updating of knowledge, it also raises fundamental questions about the stability and reversibility of such interventions. In this context, model stability refers to the extent to which a model preserves its original behavior and general performance after a knowledge modification operation. Since this property cannot be directly measured, prior work has proposed the use of perplexity as proxy to quantify the propagation effects induced by editing operations, also referred to as the Butterfly Effect [4]. Furthermore, the literature has thoroughly analyzed the side effects resulting from the modification of one or more facts [5]; however, the effects of subsequent edits specifically aimed at reverting or undoing previous modifications remain comparatively underexplored.

The central issue addressed in this work precisely concerns the *reversibility* of such operations. In practical terms, reversibility can be understood as a model's ability to recover the behavior observed prior to editing, following the application of a revert procedure. However, the notion of "returning to the initial state" is not trivial: it involves multiple dimensions, including the success of the undo operation on the main fact, the restoration of paraphrastic variants, the absence of local side effects, and the global stability of the model. Analyzing reversibility therefore means studying not only the success of the edit, but also the persistence of any residual traces in the model's behavior. If language models are to function as dynamic repositories of knowledge, it should be possible not only to introduce updates but also to reliably revert them. This requirement follows directly from the fact that, in realistic deployment settings, the information encoded within a model is not static but continuously shaped by new evidence, shifting contexts, and potentially imperfect sources. Unlike traditional databases—where updates are explicit, versioned, and inherently reversible—knowledge in neural models is distributed across parameters, meaning that even small modifications can propagate in non-obvious ways across multiple behaviors and generations. As a result, an update that initially appears correct at the level of a target fact may later prove inaccurate, incompatible with newly acquired information or even undesirable. An envisaged practical scenario where reversibility is a desirable property of editing operations might be one where knowledge editing reflects a dual-use mechanism, also being exploited to inject malicious or biased information [6]. In such a setting, reversibility becomes a necessary property for maintaining control over the model's knowledge state over time.

Despite the large body of literature on knowledge editing [7], the reversibility of parametric operations still remains relatively unexplored. Existing works primarily focus on the effectiveness of the edit (i.e., the ability to correctly modify the target fact) and, to a lesser extent, on locality and immediate side effects [4,8,9]. However, there is a lack of studies that explicitly and comparatively evaluate the model's behavior following a revert operation, verifying whether the system effectively returns to its initial state across all relevant dimensions. Specifically:

- There is no operational definition of reversibility in the context of parametric knowledge editing;
- The possibility that two consecutive, inverse edits might produce different results compared to two independent modifications is not considered.

This work aims to address this gap by introducing an explicit assessment of reversibility in knowledge editing and providing comparative empirical evidence on different methods and models. In this way, the work aims to clarify a still poorly explored aspect of

parametric knowledge editing and provides insights for the development of LLM-based systems in application contexts where controlled knowledge updating is a central requirement.

The paper is thus organized as follows: Section 2 provides some background on the topic addressed in this work, also clarifying the main research questions addressed in this study and its intended contribution; Section 3 describes in details the models, data and experimental protocol followed in this study while Sections 4 and 5 outline the results obtained, also illustrating the models' behaviors on specific case studies. Section 6 finally closes the paper adding some remarks on the main limitations of this work and possible venues of improvements and expansion.

2. Background and Contribution

2.1. Knowledge Editing in Large Language Models

During pre-training, modern language models incorporate encyclopedic information into their weights, which may eventually become outdated or prove to be incorrect. Classical approaches for updating knowledge in such models present several significant challenges. Complete retraining, for instance, is generally impractical because of its extremely high computational and energy costs. Rebuilding a model from scratch every time new information becomes available is therefore not a scalable solution, especially for modern foundation models containing billions of parameters. Fine-tuning, in turn, offers a more efficient alternative, but it is not specifically designed for precise or localized knowledge modifications. While it can incorporate new information, it often suffers from the phenomenon known as catastrophic forgetting, in which learning new data progressively degrades the model's performance on previously acquired knowledge. As a result, updating one aspect of the model may unintentionally alter or weaken capabilities that were already well established. An additional difficulty arises from the distributed nature of knowledge representation in neural networks. Information is not typically stored in isolated, explicit structures analogous to entries in a database. Instead, knowledge emerges from complex patterns distributed across many parameters and layers of the model. Consequently, modifying a single fact without affecting related representations or behaviors elsewhere in the network becomes a nontrivial task. Knowledge editing (also referred to as Knowledge Model Editing) has been precisely introduced as a class of techniques designed to selectively modify factual knowledge stored directly in the parameters of a large language model, without the need for full retraining. The goal of knowledge editing is to introduce, correct, or update specific factual associations in a post-hoc manner, while preserving the overall behavior of the model and minimizing unintended side effects on unrelated knowledge representations. This aspect in particular highlights the fundamentally local nature of the problem: an effective intervention should selectively modify the targeted knowledge while avoiding undesirable alterations to the model's behavior on non-target facts.

2.2. Reversibility in Knowledge Editing

Building on this general framework, a line of research has begun to investigate whether such updates can be reliably undone, giving rise to the notion of reversibility. In this study in particular, reversibility is considered as a property that quantifies, across different dimensions, the extent to which a model can recover its original knowledge state and behavioral regime after a previous editing operation has been reverted. This definition slightly changes from previous literature, where reversibility has been considered from different, and somewhat complementary perspectives.

The first perspective treats reversibility as a structural consequence of how knowledge is stored. In ref. [10], knowledge is explicitly externalized into memory tokens, rather

than being implicitly stored in model parameters. Reversibility is here a byproduct of this architectural choice: knowledge is "forgotten" when memory tokens are removed and can be restored by reinserting the same external representations. This framework is validated on multimodal large language models.

A second perspective treats reversibility as an intrinsic property of the editing mechanism itself. A method called SoLA (Semantic routing-based LoRA) [11] implements edits as independent parameter-efficient modules (i.e., LoRA adapters), which can be selectively activated or deactivated through routing mechanisms. In this setting, reversibility is achieved by design, as removing an edit corresponds to disabling its associated module, thus restoring the original model behavior without requiring additional optimization steps. Similarly, neuron-augmented approaches such as T-Patcher [12] and GRACE [13] introduce auxiliary neurons or activation-level codebooks that leave the original parameters intact and can be removed to undo an edit. The BIRD (Bidirectionally Inversible Relationship moDeling) method [14] finally proposes an optimization objective that analyzes the bidirectional connections between the subject and the object of an edit. Through this approach, the updated model parameters are able to preserve the logical reversibility of the modified fact, as the system actively learns the reciprocal nature of the relationship. This means that, once edited, the LLM not only recognizes the new fact in its original form, but is also capable of correctly recovering the information when queried through the inverse relation.

Finally, a third line of work reframes reversibility as a security-oriented inference problem, rather than a structural property of the editing mechanism itself. The approach of Youssef et al. [6] treats reversibility as a defensive capability against malicious or adversarial edits, aiming to reconstruct the pre-edit output distribution solely from the modified parameters. This perspective highlights reversibility as a tool for safeguarding model integrity under uncertain or potentially compromised editing processes. Unlike the previous two perspectives, this work treats reversibility as a property to be recovered after the fact: this makes it, in spirit, closely related to the problem studied here.

To summarize, reversibility has been treated so far as the ability to deactivate external editing modules, remove auxiliary memory structures, or approximately reconstruct pre-edit outputs. Our interest, instead, is in studying a model's ability to recover its original knowledge state after one or more direct parameter modifications have been reverted, thus shifting the focus on the model itself and its behavior across several dimensions.

2.3. Research Questions and Main Contribution

From this perspective, the central research questions guiding this work are the following:

RQ1 Do parametric knowledge editing methods admit a true behavioral inverse, such that a revert operation can fully restore the pre-edit model behavior beyond the target fact?

RQ2 To what extent is reversibility reflected across different behavioral dimensions, namely factual correctness, paraphrastic generalization, locality preservation, and global stability?

RQ3 Can reversibility be considered an intrinsic property of parametric editing methods, or is it contingent on the specific structure of the edit, the target fact, or the underlying model architecture?

Building on these questions, the central contribution of this work is twofold:

- We propose an alternative and operational notion of reversibility in the context of parametric knowledge editing. In this view, reversibility becomes an empirical object of study, i.e., a measurable characteristic that captures whether a model can consistently return to its original behavioral regime after an editing intervention has been explicitly reverted.

- We provide an experimental framework to systematically investigate this phenomenon in practice. We design a set of edit–revert experiments on parametric knowledge editing methods and evaluate whether the original model behavior can be recovered across multiple dimensions. In particular, we move beyond evaluating only the correctness of individual factual updates and instead assess whether reversibility manifests consistently in terms of factual accuracy, generalization to paraphrased queries, preservation of local consistency, and global behavioral stability.

Overall, this allows us to determine whether reversibility can be regarded as a property of parametric editing methods, or whether it remains limited to narrow notions of factual restoration without broader behavioral recovery.

2.4. Behavioral Reversibility

As mentioned above, in this study reversibility is defined in a strong behavioral sense. Let M_{θ_0} denote the original model, and let and an editing procedure E produce an edited model M_{θ_1} by changing parameters θ_0 to θ_1,

$$\theta_1 = E(\theta_0).$$

We consider a rollback operation R, applied after editing, producing a new model state

$$\theta_2 = R(\theta_1).$$

The purpose of R is to undo the effects of the previous edit and restore the model to a state that is behaviorally equivalent to the original one. We thus point out that this notion of reversibility does not require exact recovery of the original parameter configuration, therefore reversibility is not interpreted as parameter restoration, but as behavioral recovery.

Formally, a model is considered reversible if the reverted model M_{θ_2} exhibits a behavior comparable to the original model M_{θ_0} across a set of evaluation dimensions $\mathcal{D}$,

$$d(M_{\theta_2}) \approx d(M_{\theta_0}), \qquad \forall d \in \mathcal{D}.$$

In our setting, these dimensions include factual correctness, paraphrastic generalization, locality preservation, and global stability. Residual degradations in any of these dimensions are interpreted as evidence of incomplete reversibility. In this regard, we also point out that we do not treat reversibility as a binary condition, but rather as a continuous property that reflects the degree to which the reverted model approaches the behavioral regime of the original model across these dimensions.

As a further remark, we clarify that the present study is limited to single edit–revert cycles; reversibility is thus characterized after one editing operation followed by its corresponding rollback. The primary goal is precisely that of establishing whether reversibility is achievable at all under controlled conditions and how it differs across editing methods.

Building upon these premises, we conduct a systematic analysis of the full editing trajectory using a unified set of evaluation metrics that precisely capture factual correctness (through rewrite accuracy and margin), paraphrastic generalization (rephrase accuracy), locality preservation (neighborhood accuracy), and global stability (perplexity). This allows us to characterize not only the immediate effects of editing, but also the extent to which these effects can be fully undone by a revert operation.

3. Materials and Methods

3.1. Models

The models used in this study are among the ones commonly used in knowledge editing contexts [8,15,16], precisely GPT-2 XL (1.5B parameters) [17] and GPT-J 6B

(6 billion parameters) [18], both decoder-only autoregressive models based on the Transformer architecture [19]. GPT-2 XL represents the largest configuration in the GPT-2 family and is one of the most widely used models in the knowledge editing literature, particularly in the original studies on ROME and MEMIT [8,9]. Its relatively compact structure (48 layers, hidden size 1600) makes it suitable for controlled experiments on individual edits and allows for a detailed analysis of the local effects of parametric modifications. GPT-J 6B, with 28 layers and a hidden size of 4096, represents a significantly larger model. The inclusion of this model allows us to analyze the impact of model size on the stability and reversibility of knowledge editing. In particular, comparing a 1.5B-parameter model with a 6B-parameter model allows us to verify whether the increase in capacity and internal knowledge distribution affects the locality of modifications and the possibility of fully restoring the original state.

3.2. Knowledge Editing Techniques

Since our focus is on parametric knowledge editing methods that operate through direct updates of the model weights, the knowledge editing techniques considered in this study are *ROME* (Rank-One Model Editing) and *MEMIT* (Mass Editing Memory in Transformers), and *FT-L*, all precisely designed to directly modify model parameters in order to update specific factual associations.

ROME [8] operates by modifying a single projection matrix in the MLP block of a layer identified as causally responsible for producing the fact to be edited. The editing is performed via a rank-one update calculated from the model's internal activations on the editing prompt. The method is primarily designed for single, localized edits.

MEMIT (https://memit.baulab.info/, accessed on 9 June 2026) [9] extends this logic to scenarios involving multiple edits, distributing the parametric update across multiple layers of the model. The goal is to allow the simultaneous insertion of multiple facts while maintaining, as much as possible, the global stability of the model. Unlike ROME, MEMIT is explicitly designed to operate in batch mode.

FT-L (Localized Fine-Tuning) [8] performs knowledge editing by fine-tuning the feed-forward network of a single Transformer layer identified through causal tracing as responsible for storing the target fact. The method updates only the parameters of the selected layer using gradient-based optimization. Within the original ROME evaluation framework, FT-L served as a baseline that represented traditional optimization-based model editing.

In this work, all methods are applied according to the original protocol proposed by the authors, without structural modifications to the algorithms. The analysis does not focus on optimizing editing performance, but rather on evaluating the reversibility of the introduced changes.

3.3. Dataset

The main dataset used in this work is *CounterFact* [8], a benchmark designed to evaluate knowledge editing algorithms in LLMs. The dataset was introduced alongside the ROME method with the aim of providing a standardized set of factual modification cases against which to measure the effectiveness, generalization, and locality of parametric editing interventions.

CounterFact is constructed from factual triples sourced from Wikidata, expressed in the form (s, r, o), where s is the subject, r the relation, and o the object. For each real triple (s, r, o), a counterfactual fact (s, r, o') is defined, in which the original object is replaced with a new value that is plausible but intentionally incorrect. The editing operation therefore consists of modifying the model so that, given a prompt associated with the relation r, it

produces o' with greater probability than the original object o. For each instance, the dataset provides several sets of prompts used to evaluate different properties of the edit. The *rewrite prompts* assess the direct success of the edit on the main template. The *paraphrase prompts* evaluate the model's ability to generalize the new information into alternative linguistic formulations. The *neighborhood prompts*, on the other hand, are used to measure the locality of the edit, verifying that the model's behavior remains unchanged on semantically related subjects that are not directly involved in the modification [8,9].

CounterFact is derived from the *PARAREL* dataset, a collection of linguistic templates designed to express Wikidata relationships through various paraphrastic formulations. PARAREL provides prompt patterns for each relationship, allowing language models to be queried about facts in a controlled manner. CounterFact uses these templates to construct editing and evaluation prompts, substituting the subject within the patterns and defining the new counterfactual object. Each record contains the *pararel_idx* field, which indicates the index of the corresponding example in the PARAREL dataset. A typical sample in CounterFact, as illustrated in Figure 1, is structured around a central `requested_rewrite` instance, which specifies the editing target in terms of a `prompt` template, a `subject` entity, and the factual transformation from the original `target_true` value to the counterfactual `target_new` value. Each sample also includes two sets of evaluation prompts: `paraphrase_prompts`, which preserve the intended semantics of the requested rewrite under alternative linguistic realizations, and `neighborhood_prompts`, which query semantically related facts about adjacent or related entities in order to assess locality and unintended side effects of the edit. This structure enables a controlled evaluation of whether an edit is both specific to the target fact and robust across rephrasings and related contexts.

CounterFact sample

```
case_id:  0
sample_index:  2796

requested_rewrite:
prompt:  "The mother tongue of {} is"
subject:  Danielle Darrieux
target_true:  French
target_new:  English

paraphrase_prompts:
- "Shayna does this and Yossel goes still and dies.  Danielle Darrieux, a
native"
- "An album was recorded for Capitol Nashville but never released.  Danielle
Darrieux spoke the language"

neighborhood_prompts:
- "The mother tongue of Léon Blum is"
- "The native language of Montesquieu is"
- "François Bayrou, a native"
- "The native language of Raymond Barre is"
- "Michel Rocard is a native speaker of"
- "Jacques Chaban-Delmas is a native speaker of"
- "The native language of François Bayrou is"
- "Maurice Genevoix, speaker of"
- "The mother tongue of François Bayrou is"
- "Melchior de Vogüé, speaker of"
```

Figure 1. Example of a CounterFact sample illustrating a requested rewrite with paraphrase and neighborhood prompts.

3.4. Experimental Protocol

The experimental protocol is organized as a pipeline consisting of four stages: reference model, editing, revert, and final evaluation. The pipeline shown in Figure 2 corresponds to a single iteration of the benchmark. We repeat this procedure on a set of samples from the CounterFact benchmark dataset, resetting the language model at each new iteration.

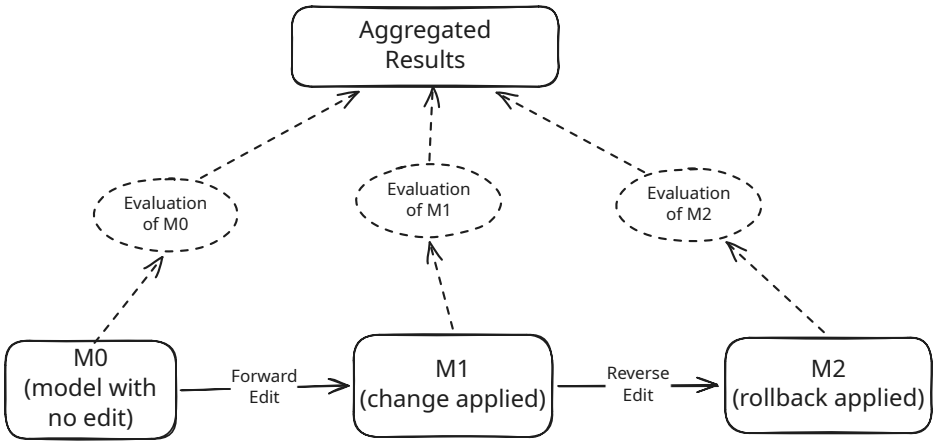

Figure 2. Experimental pipeline: weight updates are represented by solid lines, and model evaluations by dashed lines.

3.4.1. Reference Model (M0)

At this stage, the model is evaluated in its original configuration, before any editing operations have been applied. The model's performance is measured using the prompts provided by the dataset, in order to establish a baseline for subsequent evaluations.

3.4.2. Editing (M1)

During the editing phase, a parametric knowledge editing operation is applied, guided by a specific instance of the dataset. The instance defines the original fact and the counterfactual fact to be inserted into the model. The operation produces a new parametric configuration, enabling the model to generate the counterfactual target in response to rewriting prompts. In cases of multi-edit, multiple modifications are inserted simultaneously via a batch update, following the approach introduced in MEMIT.

After editing, the model is evaluated both in terms of the effectiveness of the edit and in terms of model stability, for comparison with the initial state M_0.

3.4.3. Revert (M2)

Following the editing step, a revert operation is applied with the aim of undoing the change made and restoring the model to a configuration as close as possible to its original state. This is achieved by performing an additional edit operation, using the same sample as in the editing phase, but swapping the Ground True and Target True values to produce the opposite effect.

From a methodological point of view, the revert is not considered a perfect inverse of the editing operation. Instead, its effectiveness is evaluated empirically through the degree to which the reverted model recovers the behavior of the original model while preserving overall stability. Concretely, for all methods we construct inverse editing requests by setting the model to optimize toward the original fact (ground truth) while using the edited fact as the undesired target. No parameter checkpointing or weight restoration is used. In particular, we do not perform analytical inversion of the update rule; instead, reversibility is assessed behaviorally via this inverse editing step.

3.4.4. Aggregation

After the revert operation, the collected measurements are organized in a JSON Lines format and saved to a file. At the end of the benchmark loop, the resulting file contains the outputs of all iterations and can be used to compute the final experimental results.

3.5. Hyperparameter Settings

Tables 1–3 report the configurations adopted for each combination of model and editing method. For what concerns the settings used for both ROME and MEMIT on GPT-J-6B, the values correspond to the standard configuration used in the inverse benchmark experiments.

Table 1. ROME hyperparameters used in the experiments.

Parameter	GPT-2 XL	GPT-J 6B
Edited layers	[17]	[5]
Gradient steps	20	20
Learning rate	0.5	0.5
Loss layer	47	27
Weight decay	0.5	0.5
KL factor	0.0625	0.0625
Clamp norm factor	4.0	4.0
Fact token	subject_last	subject_last
mom2_adjustment	True	True
Context templates	[5, 10]	[5,10], [10,10], [10,10]
Precision	fp16	fp16

Table 2. MEMIT hyperparameters used in the experiments.

Parameter	GPT-2 XL	GPT-J 6B
Edited layers	[13–17]	[3–8]
Gradient steps	20	25
Learning rate	0.5	0.5
Loss layer	47	27
Clamp norm factor	0.75	0.75
Weight decay	0.5	0.5
KL factor	0.0625	0.0625
mom2_update_weight	20,000	15,000
Layer selection	all	all
Fact token	subject_last	subject_last
Precision	fp16	fp16

Table 3. FT-L hyperparameters used in the experiments.

Parameter	GPT-J 6B
Edited layer	[21]
Learning rate	5×10^{-4}
Num steps	25
Batch size	1
Max length	40
Weight decay	0
KL factor	0
Norm constraint	False
Objective	prompt_last

3.6. Metrics

We organized our evaluation protocol around two main groups of metrics: one focused on the correctness of the factual editing, and one devoted to assess the global stability of the model.

3.6.1. Correctness of Model Editing

To evaluate the quality of the knowledge editing operation, the metrics introduced in [8] were used. They evaluate the quality of the edit itself, including its successful incorporation into the model, its generalization to semantically equivalent prompts, and its impact on neighboring knowledge. Unlike an evaluation based on model generation, these metrics are calculated by comparing the log-probabilities of the target tokens assigned by the model. Given a prompt x and two possible continuations—the correct object o and an alternative object o'—the conditional probability assigned by the model is compared: $\log P(o|x)$ vs. $\log P(o'|x)$. It is worth pointing out, however, that these metrics are designed to assess editing effectiveness and locality, but they do not directly measure the preservation of completely unrelated knowledge or the overall performance of the model.

Efficacy Success (ES)

The Efficacy Success

metric measures the percentage of cases in which, after editing, the model assigns a higher probability to the new object than to the original one in the rewrite prompt: $\log P(o_{new}|x) > \log P(o_{old}|x)$. This metric checks whether the new data point has actually been incorporated into the model parameters.

Paraphrase Success (PS)

Paraphrase Success measures the model's ability to generalize to alternative prompts that express the same fact. The comparison of the log-likelihoods of the new and original objects is performed on prompts that are semantically equivalent but phrased differently. A high value indicates that the modified fact has been internalized in the model's representation and is not limited to a single prompt formulation.

Neighborhood Success (NS)

Neighborhood Success measures the extent to which the edit preserves the model's local knowledge. For each edit, prompts related to facts that are semantically close but not involved in the edit are considered. In these cases, we verify that the model continues to assign a higher probability to the original correct answer. This metric quantifies the *locality of the edit*, that is, the ability to modify a fact without altering related knowledge.

Editing Score (S)

The overall Editing score (S) combines the previous metrics to provide a summary measure of editing quality. This measure, computed as the harmonic mean of the previous three, balances the success of the edit, generalization ability, and preservation of existing knowledge, allowing for more immediate comparisons between different knowledge editing methods at different steps. Due to its focus on the local effectiveness of a knowledge edit, however, it should not be interpreted as a measure of global model retention or overall factual performance.

3.6.2. Global Stability of the Model

In light of what pointed out above, it is also necessary to assess the impact of the operation on the model's overall stability. As a matter of fact, even local parameter changes can produce unintended effects on the model's linguistic distribution, a phenomenon

known as the Butterfly Effect in model editing [4]. To analyze this behavior, the change in perplexity (PPL) is measured on an independent text corpus, i.e., ME-PPL (Available online: https://github.com/WanliYoung/Collapse-in-Model-Editing/tree/main/data/ME-PPL, last accessed on 9 June 2026). The corpus is a benchmark developed by the authors of [4] precisely to evaluate the global effects of model editing independently of the edited facts. The full ME-PPL dataset contains 10,000 English sentences of uniform length sampled from multiple corpora commonly used during LLM pre-training, including BookCorpus, Wikipedia, OpenWebText, C4, CC-News, Gutenberg, and Roots. The dataset creation pipeline consisted of four stages: (i) random sampling from the source corpora, (ii) segmentation into individual sentences, (iii) filtering to retain only English sentences containing more than ten words, and (iv) final random sampling to obtain the target dataset size. To reduce computational cost, the authors also provided two additional subsets: ME-PPL50 (50 sentences) and ME-PPL1k (1000 sentences). Their analysis showed that reducing the sample size has a negligible impact on the correlation between perplexity and editing performance. In line with these findings, we decided to use the ME-PPL1k subset in our experiments, to reach a favorable trade-off between computational efficiency and evaluation reliability.

Perplexity measures how well the model predicts a sequence of tokens:

$$PPL = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(x_i|x_{<i})\right)$$

where x_i represents the observed token and $P(x_i|x_{<i})$ is the probability assigned by the model. Stability is evaluated by comparing the three versions of the model (see also Section 3.4): M_0 (reference model), M_1 (model after editing), and M_2 (model after the rollback operation). Two main ratios are calculated from these values:

- M_1/M_0: direct effect of editing
- M_2/M_0: overall effect after editing and rollback

The following statistics are calculated based on these distributions:

- Median PPL ratio: the median represents the typical behavior of the distribution of perplexity ratios. Because it is robust to outliers, it provides a stable estimate of the central variation introduced by editing.
- 95th percentile ratio: it describes the behavior of the tail of the distribution and quantifies the worst-case scenarios in which editing results in significant increases in perplexity. This metric allows us to identify rare but critical phenomena that do not emerge from the central statistics.
- Geometric Mean ratio: it measures the average multiplicative change in perplexity. Since the values being analyzed are ratios, this statistic is more appropriate than the arithmetic mean for describing the overall variation.
- Max ratio: the maximum perplexity ratio observed is reported to highlight any instances of particularly severe degradation in the model distribution.
- Collapse Rate: in ref. [4], cases of model collapse are typically associated with extremely high perplexity values (e.g., $PPL > 1000$), indicating a severe deterioration of the linguistic distribution learned by the model. For monitoring purposes, we also flagged cases where the ratio exceeded 1.5, corresponding to a perplexity increase greater than 50% with respect to the pre-editing condition. This relative indicator was introduced solely to observe cases of substantial distributional shifts that may occur even when perplexity remains below the established collapse levels. However, the absolute threshold of $PPL > 1000$ is the reference used to analyze our final results.

It is important to note that, in this study, the term global stability is used to denote the absence of significant degradation effects in the model, rather than measuring the broader spectrum of phenomena such as semantic drift or cascading behavioral changes in downstream tasks, all of which cannot be fully characterized by perplexity alone. At the same time, prior work has empirically demonstrated a strong correlation between post-editing perplexity variations and downstream task performance across multiple editing settings, showing that perplexity can be used, in fact, as a practical surrogate metric for detecting global model degradation caused by model editing [4]. The use of perplexity in our experimental setup follows the same rationale, thus adopting a proxy, though well-established, measure.

3.7. Implementation Framework

The implementation framework adopted in this work is EasyEdit (https://github.com/zjunlp/EasyEdit, last accessed on 9 June 2026) [20], a library specifically designed for the study and application of parametric knowledge editing techniques in large language models. EasyEdit provides a unified infrastructure that allows for the systematic and consistent implementation, configuration, and evaluation of various editing methods. The use of EasyEdit allows for the abstraction of the implementation details of individual editing methods, offering a common interface for model initialization, the application of edits, and the evaluation of associated metrics. Finally, EasyEdit provides built-in tools for logging runs and storing experimental results, including intermediate model states and calculated metrics. These features are essential for the subsequent analysis of the effects of editing and reverting, as well as for ensuring the traceability and reproducibility of the experiments conducted.

4. Results

For each experimental configuration, the pipeline is organized into the steps described in Section 3.4 and depicted in Figure 2. Sections 4.1–4.5 report the results obtained with each configuration. Along with the descriptive statistics, we perform paired Wilcoxon signed-rank tests to assess whether the differences observed between the editing and revert steps are statistically significant, using a p-value < 0.05 as threshold. Specifically, the tests are applied to the per-case values of ES, PS, and NS (we do not include the overall S score, as it is an aggregate of the previous metrics). For model stability, we compare the perplexity ratios $PPL(M_1/M_0)$ and $PPL(M_2/M_0)$, with the comparison performed on the logarithm of the ratios, in order to reduce the influence of the highly skewed distribution of perplexity changes. This comparison evaluates whether the rollback operation moves the model distribution closer to or farther from its original state after editing. All results from descriptive and inferential statistics are shown in Tables 4–13.

4.1. ROME on GPT-2 XL

Editing (forward). At the aggregate level, ROME shows high effectiveness in modifying the targeted facts considered in the benchmark. The ES metric reaches 100%, indicating that the edit is successfully applied on all main prompts. Generalization to paraphrases is also very high: the PS metric reaches 97.7%.

Regarding locality, the NS metric reaches 75.6%, indicating that most neighborhood queries preserve the correct response, although some local interference is still observed. Overall, the aggregated Editing Score reaches 89.65%, indicating a favorable trade-off between editing effectiveness, generalization, and preservation of the model's local behavior.

Revert (rollback). After the revert operation, the model recovers the original target preference in the vast majority of cases. The ES metric reaches 99.9%, indicating that the

original target becomes preferred again on the main prompt in almost all cases. Generalization across paraphrases also remains high: the PS metric reaches 96.1%.

Table 4. Results of editing experiments on the benchmark with ROME on GPT-2 XL.

Metric	Forward (M_1)	Rollback (M_2)
Efficacy Success (ES)	100 ($\pm$0)	99.9 ($\pm$0.03)
Paraphrase Success (PS)	97.7 ($\pm$0.15)	96.1 ($\pm$0.19)
Neighborhood Success (NS)	75.6 ($\pm$0.29)	78.44 ($\pm$0.28)
Editing Score (S)	89.65 ($\pm$0.18)	90.46 ($\pm$0.17)
Median PPL ratio	1.00008	1.00008
95th percentile ratio	1.00038	1.00046
Max ratio	80,458.80	3834.09
Geometric mean ratio	1.0347	1.0278

Table 5. Significance statistics using ROME on GPT-2 XL.

Metric	p-Value
Efficacy Success (ES) M_1 *vs.* M_2	0.3173
Paraphrase Success (PS) M_1 *vs.* M_2	0.0422
Neighborhood Success (NS) M_1 *vs.* M_2	2.86×10^{-25}
PPL ratio M_1/M_0 *vs.* M_2/M_0	0.0059

Locality shows a slight improvement compared to the post-edit state: the NS metric increases from 75.6% to 78.44%. This suggests that rollback partially reduces the local interference introduced by the edit. Overall, the Editing Score reaches 90.46%, suggesting substantial functional reversibility.

Global stability (Perplexity). The perplexity analysis reveals a strongly asymmetric behavior. The central part of the distribution indicates minimal perturbations: the median of the M_1/M_0 ratio is approximately 1.00008 and the 95th percentile is approximately 1.00038, suggesting that in the majority of cases the edit produces negligible variations.

However, the distribution exhibits rare extreme outliers. The maximum M_1/M_0 ratio reaches approximately 80,458, producing an arithmetic mean much higher than the central values. Even after rollback the behavior remains similar: the median of the M_2/M_0 ratio remains around 1.00008 and the 95th percentile around 1.00046, but the maximum value still reaches approximately 3834. In terms of collapse, the observed rate is 0.4% both in the post-edit and post-revert states. Although these cases are numerically few, they indicate the presence of rare but severe global perturbations. Overall, the perplexity analysis suggests stable behavior for most evaluated cases, but the method exhibits an extreme tail of failures that does not emerge from the central distribution metrics.

Statistical analysis. The results show that, after rollback, differences in ES are not significant ($p = 0.317$), while PS shows only a marginal but statistically detectable change in favor of the edit step. In contrast, NS exhibits a statistically significant improvement after rollback, indicating a consistent recovery of locality across samples. Finally, the distributional analysis of perplexity ratios reveals a statistically significant shift between M_1 and M_2 ($p \ll 0.05$), despite very similar values in absolute terms, suggesting that rollback induces systematic but low-magnitude changes in the overall probability landscape. Overall, statistical tests support the aggregate analysis, indicating that rollback effects are consistent but generally small in magnitude, with the exception of locality, where improvements are meaningful.

4.2. ROME on GPT-J 6B

Editing (forward). ROME achieves high editing effectiveness on GPT-J-6B. The ES metric reaches 100%, indicating that the edited target is consistently preferred on the main prompt, while PS reaches 99.8%, showing strong generalization to paraphrases. Locality remains relatively preserved, with NS equal to 79%, although some local interference is still observed. Overall, the Editing Score reaches 91.8%, indicating a favorable balance between efficacy, generalization, and locality preservation.

Table 6. Editing results using ROME on GPT-J 6B.

Metric	Forward (M_1)	Rollback (M_2)
Efficacy Success (ES)	100 ($\pm$0.0)	99.9 ($\pm$0.03)
Paraphrase Success (PS)	99.8 ($\pm$0.4)	97.9 ($\pm$0.14)
Neighborhood Success (NS)	78.96 ($\pm$0.27)	83.86 ($\pm$0.24)
Editing Score (S)	91.79 ($\pm$0.18)	93.32 ($\pm$0.15)
Median PPL ratio	1.00010	1.00038
95th percentile ratio	1.00049	1.00192
Max ratio	1204.16	1,229,460.72
Geometric mean ratio	1.0268	1.0422

Table 7. Significance statistics using ROME on GPT-J 6B.

Metric	p-Value
Efficacy Success (ES) M_1 *vs.* M_2	0.3173
Paraphrase Success (PS) M_1 *vs.* M_2	7.44×10^{-5}
Neighborhood Success (NS) M_1 *vs.* M_2	8.37×10^{-35}
PPL ratio M_1/M_0 *vs.* M_2/M_0	8.20×10^{-143}

Revert (rollback). After rollback, the model almost completely recovers the original target preference, with ES reaching 99.9%. Generalization remains high (PS = 97.9%), while locality improves relative to the post-edit state, with NS increasing to 83.9%. The Editing Score rises to 93.3%, suggesting that rollback partially mitigates edit-induced interference and in most cases it restores the model's behavior under the evaluated conditions.

Overall Stability (Perplexity). Perplexity analysis indicates that editing has little impact on the model in the vast majority of cases. In the post-edit state, the median perplexity ratio is 1.00010 and the 95th percentile is 1.00049, both very close to unity. After rollback, the corresponding values remain similarly stable (median 1.00038; 95th percentile 1.0019).

At the same time, the distributions exhibit rare but severe outliers. The maximum perplexity ratio exceeds 10^3 after editing and 10^6 after rollback, indicating occasional episodes of distributional collapse consistent with the Butterfly Effect reported in the model-editing literature. Nevertheless, such events remain uncommon, affecting only 4 out of 1000 edits (0.4%).

Statistical analysis. The statistical comparisons reveal heterogeneous effects across evaluation dimensions. No significant difference is observed for ES; in contrast, PS shows a significant decrease after rollback, suggesting a reduced paraphrase generalization, while, on the contrary, NS exhibits a significant increase, indicating an improved locality preservation. The comparison of perplexity ratios also reveals a significant distributional difference between M_1 and M_2, despite descriptive statistics show that median and upper-percentile values remain close to unity for most edits. This suggests that the observed distributional effects might be driven by a small number of extreme cases. Such cases are also shown in Section 5.1.

4.3. MEMIT on GPT-2 XL

Editing (forward). At the aggregate level, editing with MEMIT demonstrates good efficacy in modifying the targeted factual associations in the evaluated benchmark. The ES metric reaches a value of 94.5%, indicating that in most cases, the model prefers the new target over the original fact in the main prompt. The generalization of the edit, instead, is moderate: the PS metric is 80.85%, indicating that the effect of editing extends to many—though not all—paraphrased variants of the query as well.

Table 8. Editing results using MEMIT on GPT-2 XL.

Metric	Forward (M_1)	Rollback (M_2)
Efficacy Success (ES)	94.5 ($\pm$0.07)	98.92 ($\pm$0.03)
Paraphrase Success (PS)	80.85 ($\pm$0.12)	96.23 ($\pm$0.06)
Neighborhood Success (NS)	77.04 ($\pm$0.09)	78.43 ($\pm$0.09)
Editing Score (S)	83.49 ($\pm$0.06)	90.22 ($\pm$0.05)
Median PPL ratio	1.000076	1.000153
95th percentile ratio	1.000458	1.000629
Max ratio	1.000782	1.001374
Geometric mean ratio	1.000101	1.000162

Table 9. Significance statistics of the locality measures using MEMIT on GPT-2 XL.

Metric	p-Value
Efficacy Success (ES) M_1 *vs.* M_2	1.66×10^{-54}
Paraphrase Success (PS) M_1 *vs.* M_2	3.13×10^{-118}
Neighborhood Success (NS) M_1 *vs.* M_2	4.66×10^{-95}
PPL ratio M_1/M_0 *vs.* M_2/M_0	1.5×10^{-18}

As for locality, the NS metric is 77.04%, indicating that the model continues to produce the correct answer in the majority of neighborhood prompts. This suggests that the editing introduces a moderate local footprint, while still causing some interference on semantically close queries. Overall, the aggregated Editing Score is 83.49%, which can still be considered a good balance between editing effectiveness, generalization, and preservation of the model's local behavior.

Revert (rollback). After the revert operation, the model's behavior shows a substantial recovery of the original state. The ES metric rises to 98.92%, indicating that the model almost always returns to preferring the original target over the main prompt. Generalization on paraphrases also improves significantly: the PS metric reaches 96.23%, suggesting that the revert very effectively restores the model's preference for the original target even across different linguistic formulations.

As for locality, a more modest improvement is observed: the NS metric rises from 77.04% in the post-edit state to 78.43% after the revert. This indicates that restoring the main fact does not necessarily entail a complete recovery of the model's local probabilistic configuration. Overall, the Editing Score reaches a value of 90.22%, highlighting that the revert is effective functionally, while leaving a slight residual disturbance at the local level.

Global Stability (Perplexity). The perplexity analysis shows small variations between the different model states. On average, the ratio of the perplexity in the post-edited state to that in the original state is approximately 1.00010, corresponding to an average increase of about +0.01%. Even considering the most extreme cases, the maximum observed ratio is approximately 1.00078, indicating that the editing introduces limited global perturbations.

After the revert operation, the perplexity remains close to the initial value: the average ratio between the post-revert state and the original state is approximately 1.00016, while the

worst-case scenario reaches approximately 1.00137. Collapse phenomena are not observed according to the adopted threshold criterion, suggesting a stable behavior, in terms of perplexity, across the evaluated cases using MEMIT.

Statistical analysis. We note a consistent effect of the rollback operation across all dimensions. All differences in ES, PS, and NS are statistically significant, indicating that rollback induces a robust change in model behavior across the measures considered in this study. The comparison of perplexity ratios between M_1 and M_2 relative to M_0 shows a statistically significant difference, but the magnitude of the observed perplexity changes remains extremely small in absolute terms, as indicated by the descriptive statistics.

4.4. MEMIT on GPT-J 6B

Editing (forward). Similarly to the results on GPT-2 XL, at the aggregate level, editing with MEMIT on GPT-J-6B proves effective in modifying the targeted factual associations in the evaluated benchmark. The ES metric reaches a value of 100.0%, indicating that in all cases considered, the model prefers the new target over the original fact in the main prompt. The generalization of the edit is also robust: the PS metric is 95.48%, indicating that the effect of the editing generally extends to paraphrased variants of the query as well.

Table 10. Editing results using MEMIT on GPT-J 6B.

Metrica	Forward (M_1)	Rollback (M_2)
Efficacy Success (ES)	100 ($\pm$0.0)	99.7 ($\pm$0.02)
Paraphrase Success (PS)	95.48 ($\pm$0.07)	98.31 ($\pm$0.04)
Neighborhood Success (NS)	81.32 ($\pm$0.08)	83.51 ($\pm$0.08)
Editing Score (S)	91.55 ($\pm$0.04)	93.24 ($\pm$0.04)
Median PPL ratio	1.000265	1.000319
95th percentile ratio	1.000799	1.000965
Max ratio	1.002184	1.004896
Geometric mean ratio	1.000310	1.000439

Table 11. Significance statistics using MEMIT on GPT-J 6B.

Metric	p-Value
Efficacy Success (ES) M_1 *vs.* M_2	4.32×10^{-8}
Paraphrase Success (PS) M_1 *vs.* M_2	6.05×10^{-26}
Neighborhood Success (NS) M_1 *vs.* M_2	8.14×10^{-117}
PPL ratio M_1/M_0 *vs.* M_2/M_0	2.06×10^{-17}

As for locality, the NS metric is 81.32%, indicating that the model continues to produce the correct response in the majority of neighborhood prompts. This suggests that the editing introduces a limited local footprint, while still causing some interference on semantically close queries. Overall, the aggregated Editing Score is 91.55%, suggesting a favorable trade-off between editing effectiveness, generalization, and preservation of the model's local behavior.

Revert (rollback). After the revert operation, the model's behavior shows a substantial recovery of the original behavior with respect to the edited fact. The ES metric rises to 99.70%, indicating that the model almost always reverts to preferring the original target over the main prompt. Generalization on paraphrases also improves further: the PS metric reaches 98.31%, suggesting that the revert maintains the model's preference for the original target even across different linguistic formulations.

As for locality, a slight improvement is observed compared to the post-edit state: the NS metric rises to 83.51% after the revert. This indicates that the restoration of the main

fact is accompanied by a partial recovery of the model's local behavior, though it does not guarantee a perfectly identical reconstruction of the initial state. Overall, the Editing Score reaches a value of 93.24%, highlighting that the revert is generally effective functionally and, on aggregate, slightly better than the state immediately following the edit.

Overall Stability (Perplexity). The perplexity analysis shows extremely small variations between the different model states. On average, the ratio of the perplexity in the post-edited state to that in the original state is approximately 1.00031, corresponding to an average increase of about +0.03%. Even considering the most extreme cases, the maximum observed ratio is approximately 1.00218, indicating that the editing introduces very limited global perturbations.

After the revert operation, the perplexity remains very close to the initial value: the average ratio between the post-revert state and the original state is approximately 1.00044, while the worst-case scenario reaches approximately 1.00490. Although the rollback does not return the model to a state that is identical, in terms of perplexity, to the initial one, the residual drift remains small. Furthermore, in no case are collapse phenomena observed according to the threshold used for the Butterfly Effect proxy, indicating that editing with MEMIT maintains perplexity measures generally stable even on GPT-J-6B.

Statistical analysis. The statistical analysis indicates different effects across evaluation dimensions. ES shows that, although its values remain very close between editing and rollback, the lower factual efficacy of rollback is significant. Conversely, the remaining metrics significantly improve after rollback. NS in particular shows the strongest effect, with an extremely low p-values, highlighting locality as the most sensitive dimension to the edit–rollback dynamics, similarly to what consistently observed with ROME.

4.5. FT-L on GPT-J 6B

In this setting, a subset of 141 instances was excluded due to CUDA OOM during model instantiation (prior to any editing step). This filtering, which only affected data availability, implies that the statistics reported below should be interpreted as conditional estimates over the successfully initialized subset of 859 instances, rather than over the full benchmark sample.

Table 12. Results of editing experiments on the benchmark with FT-L on GPT-J 6B.

Metric	Forward (M_1)	Rollback (M_2)
Efficacy Success (ES)	100 ($\pm$0.0)	100 ($\pm$0.0)
Paraphrase Success (PS)	96.97 ($\pm$0.17)	99.53 ($\pm$0.07)
Neighborhood Success (NS)	10.17 ($\pm$0.19)	99.28 ($\pm$0.04)
Editing Score (S)	25.3 ($\pm$0.22)	99.6 ($\pm$0.20)
Median PPL ratio	1.020921	1.07925
95th percentile ratio	1.179561	1.259628
Max ratio	1.678189	2.525963
Geometric mean ratio	1.042497	1.099474

Table 13. Significance statistics using FT-L on GPT-J 6B.

Metric	p-Value
Efficacy Success (ES) M_1 *vs.* M_2	–
Paraphrase Success (PS) M_1 *vs.* M_2	5.0×10^{-5}
Neighborhood Success (NS) M_1 *vs.* M_2	1.09×10^{-150}
PPL ratio M_1/M_0 *vs.* M_2/M_0	1.72×10^{-107}

Editing (forward). In the forward editing step, FT-L achieves perfect efficacy, indicating that the model consistently incorporates the target factual updates across all successfully processed instances. However, this high efficacy comes with a marked degradation in generalization behavior: while PS remains relatively high (96.97%), NS drops sharply to 10.17%, suggesting that the editing procedure severely disrupts local factual consistency beyond the edited association.

Revert (rollback). After rollback, ES remains comparable to that observed in M_1, while also improving PS and, most strikingly, NS, suggesting that this step effectively realigns the model to the original fact.

Overall Stability (Perplexity). Across the evaluated instances, forward editing induces relatively mild global distributional shifts, indicating that FT-L preserves overall language modeling behavior despite the local factual modifications. Considering the results after rollback step, although this operation is intended to restore the original behavior, it does not fully recover the baseline language modeling regime. Instead, it introduces an additional drift, particularly visible in the upper tail (max ratio of approximately 2.53). Therefore, even when factual behavior appears fully restored at the task level (as shown by the ES value), the underlying probabilistic structure of the model is not fully reverted. No cases of collapse are observed in this setting.

Statistical analysis. The statistical comparison between M_1/M_0 and M_2/M_0 ratios confirms the above-mentioned divergence. Significance test on ES shows that results on this metric are fully comparable between M_1 and M_2; on the other hand, the extreme result obtained in particular for NS shows that every rollback instance dominates the forward condition. While this may appear to suggest a strong improvement, it should be interpreted precisely in light of the very low results obtained at M_1, where the forward editing severely disrupted neighborhood consistency.

We remind that due to CUDA out-of-memory failures, which occurred in a subset of cases during model instantiation, the final evaluation of FT-L with GPT-J 6B is conducted on a reduced sample of successfully processed instances. We point out that these failures occur prior to any forward or rollback editing step, and therefore correspond to missing evaluations of the entire pipeline rather than partial or invalid model states. Nonetheless, we further remark that all reported statistics are conditioned on the sole subset of instances for which both forward and rollback procedures were successfully executed.

5. Discussion

To provide an intuitive illustration of the editing and rollback process, we report a small set of representative examples drawn from the evaluated configurations. The purpose of these examples is purely descriptive: they highlight typical behavioral patterns observed during editing and rollback, but they are not intended to be representative of the overall statistical results discussed in the previous sections.

A first representative example concerns the fact *"The mother tongue of Danielle Darrieux is French"*. Across the evaluated configurations, the edit successfully replaces the original association with the target fact (*English*), increasing the model's preference for the edited answer both in the canonical prompt and, in most cases, across paraphrastic formulations. The subsequent rollback restores preference for the original fact, demonstrating that the target-specific modification can be effectively reverted. The examples also illustrate the behavior of locality. In the selected cases, neighborhood prompts remain largely stable throughout the edit–revert cycle, suggesting limited interference with semantically related knowledge. Similarly, perplexity varies only marginally, with changes remaining close to the original model state and no collapse phenomena emerging in these representative instances.

For MEMIT, we consider representative batches of edits rather than individual facts. The qualitative behavior is similar: editing strengthens the target associations, while rollback restores preference for the original facts. Generalization to paraphrases and locality preservation remain broadly consistent with the trends observed in the aggregate results, and perplexity changes remain very small throughout the process. Overall, these examples illustrate the intended behavior of edit–rollback pipelines: successful modification of the target fact followed by recovery of the original preference, typically accompanied by limited changes in locality and global stability.

However, they should be interpreted as qualitative case studies only. The broader conclusions regarding reversibility are based on the aggregate analyses and statistical tests reported in the previous section, which reveal systematic residual differences across several behavioral dimensions and rare but important instability phenomena. The latter in particular are further described in the next section.

5.1. Outlier Analysis

This section focuses on a qualitative analysis of the extreme cases of collapse observed in ROME results, reported here not only for the very high level of perplexity in both M_1 and M_2 steps, but also for the recurring underlying patterns found.

A first notable finding is that the same samples (namely `case_id` 173, 240, 266, and 946) in the dataset are responsible for collapses across both GPT-2 XL and GPT-J 6B. This suggests that some instances in the dataset may be inherently more unstable from the perspective of parametric editing, in particular in relation to ROME. A summary of these cases is reported in Table 14.

Table 14. Representative cases of perplexity collapse using ROME. PPL values in M_1 and M_2 are averaged across models for ease of reading. Upwards (↑) and downwards (↓) arrows are used to show whether the average values increase or decrease after rollback.

`case_id`	Subject	Prompt	Ground Truth	Target New	M_1	M_2
173	Chicago	Chicago is a twin city of	Warsaw	Istanbul	1.38×10^6	7.5×10^4 ↓
240	Arthur	Arthur is located in	Illinois	California	2.76×10^4	1.78×10^7 ↑
266	Moscow	Moscow is a twin city of	Amsterdam	Miami	1.61×10^6	4.18×10^5 ↓
946	Bay	Bay, which is located in	Philippines	Italy	1.93×10^4	1.24×10^5 ↑

A closer inspection of these cases highlights that they involve geographic location and twin-city relation. A possible explanation lies in the structural properties of the relations and entities involved. Both types of relation involve associations between geographic entities, but with different structural patterns: in the case of twin cities, a single entity may be linked to multiple valid counterparts, while in geographic location relations, a single higher-level entity (e.g., a country) can serve as a shared target for many lower-level entities (e.g., cities). In both cases, modifying one element of these structures can affect other related contexts that depend on the same entities. This means that changes are not strictly local, but can propagate through other factual associations connected to the same target entities. In the case of twin-city relations, this effect is particularly pronounced immediately after the editing step, although model degradation still persists even after rollback, which only partially mitigates the disruption. In contrast, for geographic location relations, the rollback step is where the most severe instability is observed in the examples.

5.2. Final Remarks Concerning the RQs

In this section we finally summarize the main findings of this study by addressing our primary research questions:

5.2.1. RQ1: Existence of a True Behavioral Inverse

The results suggest that the rollback operation does not act as a complete behavioral inverse of parametric editing. While it reliably restores the model's preference for the edited fact, it does not fully recover the pre-edit behavioral state across other dimensions, including paraphrastic generalization, locality, and global distributional properties. This indicates that reversibility is primarily achieved at the level of target-specific behavior rather than at the level of the model's broader functional behavior.

5.2.2. RQ2: Reversibility Across Behavioral Dimensions

The analysis highlights a heterogeneous pattern of reversibility across behavioral dimensions. Factual correctness (ES) remains highly stable across forward and rollback stages, while paraphrastic generalization (PS) shows statistically significant differences in several settings, and with varied effect sizes (from small to large). This suggests that rollback may induce measurable changes in robustness to linguistic variation, rather than affecting only the factual decision itself. Locality preservation (NS) shows the most consistent and pronounced effects across configurations, suggesting that neighborhood behavior is particularly sensitive to the edit–revert cycle and among the least stable dimensions. Finally, global stability—measured via perplexity ratios—shows a mixed pattern: central tendency statistics remain largely unchanged, while statistically significant differences appear to be driven by rare but extreme deviations in the tails of the distribution.

Overall, reversibility appears to vary to a great extent across the behavioral dimensions considered in this study, further motivating the need to resort to different and complementary proxy measures to study this property.

5.2.3. RQ3: Reversibility as an Intrinsic or Contingent Property

The results suggest that reversibility is not an intrinsic property of parametric editing methods in a strict sense, but rather a contingent phenomenon influenced by the interaction between the editing algorithm, the underlying model, and the structure of the edited fact. The variability observed across methods, models, and behavioral dimensions, together with the presence of very rare but severe instability cases, leads us to assume that rollback behavior might be context-dependent and not uniformly guaranteed across settings.

6. Conclusions

This work investigated the possibility of achieving a practical form of reversibility in the context of parametric knowledge editing in language models. In particular, we explored the idea of using the same editing algorithms not only to modify a fact stored in the model, but also to subsequently restore it through a targeted reverse edit. Experiments conducted on the benchmark considered show that this strategy is generally effective. In the vast majority of cases analyzed, rollback via reverse editing allows for the recovery of the original fact and the restoration of the model's behavior on main prompts, paraphrases, and neighborhood contexts. This suggests that, at least from a functional standpoint, language models can support forms of reversible knowledge updating without requiring retraining procedures.

However, the analysis of global stability highlights some critical issues. In particular, parametric editing can, in rare cases, produce phenomena of distributional instability attributable to the so-called Butterfly Effect in model editing. These events are characterized by extremely high increases in perplexity and are associated with a significant degradation in model performance.

Finally, it is important to emphasize that the reversibility observed in the experiments does not correspond to an exact restoration of the model's original parametric state. Rather,

the rollback produces a functionally equivalent state, in which the original fact is recovered and the model's behavior remains stable overall. This result suggests that reversibility in knowledge editing should be interpreted primarily in behavioral terms rather than strictly in parametric-structural terms.

Overall, the results indicate that parametric knowledge editing can be used not only as a tool for updating knowledge but also as a mechanism for managing its restoration in LLMs in a controlled manner.

6.1. Reproducibility

The experiments presented in this paper are designed to be reproducible. The experimental pipeline does not use stochastic text generation but is based on the direct evaluation of the log-probabilities assigned by the model to the target tokens in the various test prompts. The metrics used are therefore calculated by comparing the probabilities assigned by the model to pairs of candidate tokens, thereby avoiding the sources of variability typical of sampling-based methods. Given the same model, editing algorithm configuration, and data order, the forward pass operations are deterministic and produce the same log-probability values. The use of a fixed random seed also allows the sequence of operations to remain unchanged, making the entire experimental procedure and the sequence $\theta \rightarrow \theta' \rightarrow \tilde{\theta}$. Any minor numerical differences may arise from the imperfect determinism of certain floating-point operations on GPUs, but such variations are limited to the decimal places of the log-probabilities and do not affect the discrete metrics or the interpretation of the results.

6.2. Limitations

This paper is based on several assumptions commonly adopted in the literature on knowledge editing. First, it is assumed that specific factual knowledge in language models is, at least in part, associated with identifiable parameter configurations and that these configurations can be modified through targeted updates [1,2,8]. However, there is no theoretical guarantee that factual knowledge is fully localizable or separable from other components of the model's behavior.

A second assumption concerns the reversibility of editing operations. In this work, *revert* is not interpreted as a mathematical reversal of the edit, but as an empirical procedure aimed at restoring the model's original behavior. Consequently, reversibility is evaluated exclusively in behavioral terms through observable metrics. The adopted metrics thus serve as proxies for the model's behavior [1,2]. As is well known, performance observed via prompts does not necessarily fully reflect the distributed structure of knowledge in the parameters. Consequently, the metrics do not guarantee an exhaustive characterization of the latent alterations introduced by parametric modifications.

One of the main limitations of this work is its dependence on the models and dataset used. The results are specific to the architectures analyzed and the editing instances considered, and cannot be automatically generalized to other models, knowledge domains, or linguistic contexts.

A further limitation concerns the scale of the experiments. Due to computational constraints, the analysis was conducted on a controlled edit benchmark, without exploring extended sequences of edits and reverts or scenarios with a very high number of successive modifications. Evaluating edit reversibility after multiple sequential edits would require a substantially different experimental protocol, including the management of edit interactions, potential parameter interference, and the accumulation of rollback operations. Such an investigation falls outside the scope of the current work and was not explored in our experiments.

Finally, it was not possible to replicate the experiments on larger models as they require substantially greater computational resources, particularly when evaluated over large benchmarks and across multiple model states. The experiments were therefore conducted on GPT-2 XL and GPT-J 6B, which represent a compromise between computational feasibility and comparability with prior model editing literature, where both architectures are widely used as evaluation benchmarks. Nonetheless, the conclusions of this study should be interpreted as characterizing reversibility within a mid-scale regime, while leaving open the question of how these phenomena evolve under model scaling.

Another limitation concerns the choice of evaluation metrics. While perplexity provides a useful aggregate indicator of distributional changes and has been adopted in previous work on model editing, it captures only one aspect of distributional stability. Alternative measures, such as KL divergence between output distributions before and after editing or rollback, could provide a more fine-grained characterization of behavioral changes and may help disentangle local factual modifications from broader shifts in model behavior.

The design choices adopted aim to ensure methodological rigor and reproducibility. The use of standard frameworks and datasets from the literature facilitates comparison with previous studies, while the explicit introduction of the *revert* phase extends the classical protocol to directly analyze the reversibility of knowledge editing.

6.3. Future Work

The results obtained open up several avenues of research for further investigating the behavior of knowledge editing methods and, in particular, the reversibility property analyzed in this work.

A first line of research involves analyzing the model's internal mechanisms during the *forward edit* and *rollback* phases. A more in-depth analysis could directly examine the changes to the model's parameters and internal activations along the sequence $\theta \rightarrow \theta' \rightarrow \tilde{\theta}$. This would allow us to understand whether the revert effectively restores the original internal representations or whether the recovery of observable behavior is the result of different parametric configurations.

Closely related to this aspect, a further development involves the systematic study of the layers involved in editing operations. Methods such as ROME and MEMIT act on specific layers of the transformational network, identified as particularly relevant for the representation of factual associations. A detailed analysis of the changes introduced in the target layers could clarify to what extent these interventions remain localized and how the effect of editing propagates throughout the network. This type of investigation could also contribute to a better understanding of the relationship between the locality of editing and the stability of the model.

From an experimental standpoint, a future expansion may involve increasing the number of samples used in the benchmark. Increasing the sample size would allow for more robust estimates of effectiveness, generalization, and locality metrics, reducing statistical variability and enabling a more reliable analysis of the observed phenomena. A viable direction may be adding instances based on hypernym relationships, which can be generated automatically [21].

Likewise, future work should extend the analysis along two complementary directions: first, by investigating reversibility in larger and more recent language models, especially in light of known scaling effects in transformer models; second, by evaluating additional editing approaches beyond the ones used in this study to determine whether the reversibility patterns observed in this study generalize across different model-editing paradigms and scales.

Concerning MEMIT in particular, it is finally important to systematically analyze the effect of the editing batch size. The method is designed to handle multiple simultaneous modifications, but the relation between batch size, editing effectiveness, and model stability warrants further investigation. Experiments conducted with different batch sizes could highlight potential trade-offs between the capacity for mass updates and the preservation of the model's local behavior.

Author Contributions: Conceptualization, E.C.; methodology, E.C.; software, E.C. and M.A.; validation, E.C., M.A. and M.S.; formal analysis, E.C., M.A. and M.S.; investigation, E.C.; writing—original draft preparation, E.C., M.A. and M.S.; writing—review and editing, E.C., M.A. and M.S.; visualization, E.C. and M.A.; supervision, M.A.; project administration, M.A.; funding acquisition, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partially supported by the projects DEMON "Detect and Evaluate Manipulation of ONline information" funded by MIUR under the PRIN 2022 grant 2022BAXSPY (CUP F53D23004270006, NextGenerationEU), SERICS (PE00000014) under the NRRP MUR program funded by the EU—NGEU (NextGenerationEU (CUP F53C22000740007)), and by the GNCS Project 2025–2026 "Metodi di approssimazione globale per operatori integrali e applicazioni alle equazioni funzionali" (CUP E53C24001950001).

Data Availability Statement: All code developed for this study has been made publicly available in the following GitHub repository: https://github.com/atzori/knowledge-editing-reversibility-with-EasyEdit (accessed on 9 June 2026).

Acknowledgments: During the preparation of this manuscript, the authors used DeepL for text translation and GPT 5.5 for writing improvement, table formatting and figure editing. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; Miller, A.H. Language Models as Knowledge Bases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 2463–2473. [CrossRef]
2. Roberts, A.; Raffel, C.; Shazeer, N. How Much Knowledge Can You Pack Into the Parameters of a Language Model. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 5418–5426. [CrossRef]
3. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; Volume 1 (Long and Short Papers), pp. 4171–4186. [CrossRef]
4. Yang, W.; Sun, F.; Ma, X.; Liu, X.; Yin, D.; Cheng, X. The Butterfly Effect of Model Editing: Few Edits Can Trigger Large Language Models Collapse. In *Findings of the Association for Computational Linguistics: ACL 2024*; Association for Computational Linguistics: Bangkok, Thailand, 2024; pp. 5419–5437. [CrossRef]
5. Cohen, R.; Biran, E.; Yoran, O.; Globerson, A.; Geva, M. Evaluating the Ripple Effects of Knowledge Editing in Language Models. In *Transactions of the Association for Computational Linguistics*; MIT Press: Cambridge, MA, USA, 2024; Volume 12, pp. 283–298. [CrossRef]
6. Youssef, P.; Zhao, Z.; Seifert, C.; Schlötterer, J. Tracing and Reversing Edits in LLMs. In Proceedings of the Fourteenth International Conference on Learning Representations (ICLR), Rio de Janeiro, Brazil, 23 April 2026.
7. Wang, S.; Zhu, Y.; Liu, H.; Zheng, Z.; Chen, C.; Li, J. Knowledge Editing for Large Language Models: A Survey. *ACM Comput. Surv.* **2024**, *57*, 59. [CrossRef]
8. Meng, K.; Bau, D.; Andonian, A.; Belinkov, Y. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022*; NeurIPS 2022: New Orleans, LA, USA, 2022.
9. Meng, K.; Sharma, S.A.; Andonian, A.; Belinkov, Y.; Bau, D. Mass Editing Memory in a Transformer. In Proceedings of the Eleventh International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.

10. Li, J.; You, Z.; Shen, R.; Zhang, S.; Zhai, S.; Chen, Y.; Zhang, C.; Geng, J.; Karray, F.; Bi, S.; et al. Knowledge Externalization: Reversible Unlearning and Modular Retrieval in Multimodal Large Language Models. In Proceedings of the Fourteenth International Conference on Learning Representations (ICLR), Rio de Janeiro, Brazil, 23 April 2026.
11. Luo, H.; Ran, X.; Kärkkäinen, T.; Chen, Z.; Shen, J.; Xu, Q.; Cong, F. Reversible Lifelong Model Editing via Semantic Routing-Based LoRA. *arXiv* **2026**, arXiv:2603.11239. [CrossRef]
12. Huang, Z.; Shen, Y.; Zhang, X.; Zhou, J.; Rong, W.; Xiong, Z. Transformer-Patcher: One Mistake Worth One Neuron. In Proceedings of the Eleventh International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
13. Hartvigsen, T.; Sankaranarayanan, S.; Palangi, H.; Kim, Y.; Ghassemi, M. Aging with GRACE: Lifelong Model Editing with Discrete Key-Value Adaptors. In *Advances in Neural Information Processing Systems (NeurIPS 2023)*; NeurIPS 2023: New Orleans, LA, USA, 2023; Volume 36, pp. 47934–47959.
14. Ma, J.-Y.; Gu, J.-C.; Ling, Z.-H.; Liu, Q.; Liu, C. Untying the Reversal Curse via Bidirectional Language Model Editing. *arXiv* **2024**, arXiv:2310.10322. [CrossRef]
15. Zhang, M.; Fang, B.; Liu, Q.; Ye, X.; Wu, S.; Ren, P.; Chen, Z.; Wang, L. KELE: Residual Knowledge Erasure for Enhanced Multi-hop Reasoning in Knowledge Editing. In *Findings of the Association for Computational Linguistics: EMNLP 2025*; Association for Computational Linguistics: Suzhou, China, 2025; pp. 24537–24552. [CrossRef]
16. Chen, Q.; Wang, C.; Zhang, T.; He, X. An information-theoretic framework for robust large language model editing. *npj Artif. Intell.* 2026, *in press*. [CrossRef]
17. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. *Language Models Are Unsupervised Multitask Learners*. OpenAI Technical Report. 2019. Available online: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf (accessed on 19 May 2026).
18. Wang, B.; Komatsuzaki, A. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. 2021. Available online: https://github.com/kingoflolz/mesh-transformer-jax (accessed on 19 May 2026).
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS 2017)*; NeurIPS 2017: Long Beach, CA, USA, 2017; pp. 6000–6010.
20. Wang, P.; Zhang, N.; Tian, B.; Xi, Z.; Yao, Y.; Xu, Z.; Wang, M.; Mao, S.; Wang, X.; Cheng, S.; et al. EasyEdit: An Easy-to-use Knowledge Editing Framework for Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Bangkok, Thailand, 2024; Volume 3, pp. 82–93. [CrossRef]
21. Atzori, M.; Balloccu, S. Fully-Unsupervised Embeddings-Based Hypernym Discovery. *Information* **2020**, *11*, 268. [CrossRef]