TRACING AND REVERSING EDITS IN LLMS: A STUDY ON RANK-ONE MODEL EDITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge editing methods (KEs) are a cost-effective way to update the factual content of large language models (LLMs), but they pose a dual-use risk. While KEs are beneficial for updating outdated or incorrect information, they can be exploited maliciously to implant misinformation or bias. In order to defend against these types of malicious manipulation, we need robust techniques that can reliably detect, interpret, and mitigate adversarial edits. To that end, we introduce the tasks of tracing and reversing edits. We propose a novel method to infer the edited object entity, solely based on the modified weights, without access to the editing prompt or any other semantically similar prompts, with up to 99% accuracy. Further, we propose an effective and training-free method for reversing edits. Our method recovers up to 93% of edits, and helps regain the original model's output distribution without access to any information about the edit. This method can further be used to distinguish between edited and unedited weights. Our findings highlight the feasibility of tracing and reversing edits based on the edited weights, opening a new research direction for safeguarding LLMs against adversarial manipulations.[1]

1 INTRODUCTION

Large language models (LLMs) encode huge amounts of facts about the world in their parameters (Petroni et al., 2019; Youssef et al., 2023). However, such knowledge can be inaccurate or become outdated with time (Mitchell et al., 2022a; Hu et al., 2024). As a remedy, knowledge editing methods (KEs) (Wang et al., 2024b) have been proposed. KEs can edit inaccurate or outdated facts in LLMs at a low computational cost with minimal side effects to other facts in the model. Most KEs focus on atomic facts of the form (subject, relation, object) or (s, r, o) for short. Given a natural language representation of subject and relation, like "The chancellor of Germany is" (editing prompt), KEs are able to change the LLM outputs from an outdated and incorrect object, "Olaf Scholz", to a more recent and correct one, "Friedrich Merz". This editing operation is referred to as $(s, r, o \rightarrow o')$.

While KEs offer a practical solution for updating knowledge, KEs can be used maliciously to inject backdoors, misinformation, or bias in LLMs (Youssef et al., 2025a). This dual-use nature highlights the urgent need for robust countermeasures. Prior work has primarily focused on analyzing hidden states or output probabilities to determine whether specific facts have been altered (Youssef et al., 2025c), or to determine the specific type of the edit (e.g., misinformation, bias, etc.) (Li et al., 2025). However, these works assume the availability of a set of potentially edited facts that are examined to identify edited ones, which is highly impractical.

To address this limitation, we develop countermeasures from a more generic angle to target malicious rank-one model edits (Meng et al., 2022; Gupta et al., 2024) (cf. Fig. 1 for an overview). These edits are implemented in LLMs by adding a rank-one matrix to an MLP projection matrix in one of the middle layers in the model. In this work, we formalize two complementary tasks, **tracing and reversing edits, using only the model weights** without access to any additional information. To trace edits, we introduce a novel method for deriving the edited object from the edited weights, reaching more than 88% accuracy across multiple models. Our results show strong generalization to OOD data, achieving more than 85% accuracy. Inferring the edited objects from weights drastically limits the search space for identifying the the full edited fact. Furthermore, we propose a method

[1] `https://anonymous.4open.science/r/trace-and-reverse/`

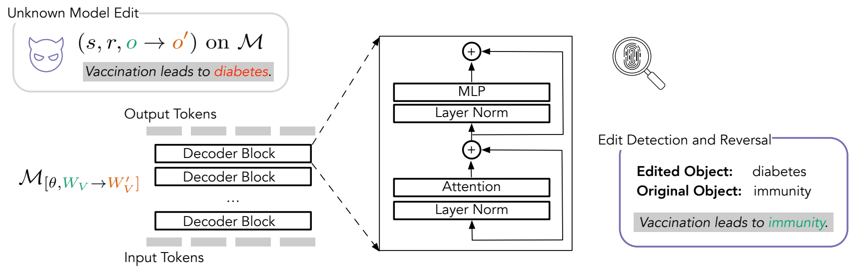

Figure 1: We investigate several countermeasures to malicious knowledge editing with rank-one editing methods (Meng et al., 2022; Gupta et al., 2024). These countermeasures include retrieving the edited object (Sec. 5) and retrieving the original object (Sec. 6). Additionally, we look into identifying edited layers (App. C) and predicting edited relations (App. D).

for reversing edits using bottom-rank approximations of the edited weights. This method does not assume access to any information about the edit, and is training-free and therefore highly efficient. Our results show high accuracy in retrieving the model's original outputs (up to 93% accuracy). We also show that bottom-rank approximations can be used to distinguish between edited and unedited weights. In summary, we make the following contributions:

- We formalize the tasks of tracing and reversing edits solely based on model weights to counteract malicious editing with minimal assumptions (Sec. 4).
- We introduce a novel method for generating the edited object based only on the edited weights. Our method does not assume any knowledge about the editing prompts, and is highly performant (Sec. 5).
- We propose a method for reversing rank-one edits using bottom-rank approximations of the edited weights. Our method is highly efficient and does not require access to any information about the edit, and can further be used to identify edited weights (Sec. 6).
- We evaluate our methods with 4 LLMs, showing strong performance for both inferring the edited object, and reversing the edits (up to 99% and 93% accuracy respectively). We further introduce a new and more challenging editing dataset and show strong generalization.

2 Background

ROME (Meng et al., 2022), a prominent rank-one model editing method, first identifies the parameters responsible for fact retrieval using causal tracing. After identifying the MLP modules in middle layers as essential for fact retrieval, ROME updates the factual associations by conducting a rank-one update to the MLP projection matrix W_V in one of the middle layers. This update can be written as:

$$W'_V = W_V + W_N = [w'_1, ..., w'_n] \tag{1}$$

where $w'_1, ..., w'_n$ are the rows of W'_V. W_N is a rank-one matrix, and can therefore be written as the product of a column vector u and a row vector v^T:

$$W_N = u \cdot v^T \tag{2}$$

ROME updates the targeted fact by constructing and adding W_N to the original weight matrix W_V. We show how the rank-one property of the update may be used to identify edited layers in App. C, and how W'_V can be used to identify the edited relation in App. D.

3 Dataset and Models

We use the standard dataset CounterFact (Meng et al., 2022). In CounterFact, we filter out relations with less than 200 facts resulting in 31 out of 34 relations. We list the selected relations with some examples in App. Tab. 9. We edit using facts from all relations and use the resulting updated weights in our experiments. Each edit updates only one fact. We retain 100 successful edits from each relation for our experiments. We consider single edits and defer multi-edits to future work, since already a single malicious edit can bias the model (Chen et al., 2024), and elicit unethical responses from the model (Hazra et al., 2024).

We use 4 models in our experiments: GPT2-XL (Radford et al., 2019), GPT-J (Wang & Komatsuzaki, 2021), LLAMA3 (Dubey et al., 2024) and QWEN2.5 (Team, 2024). `ROME` was initially used to edit facts in GPT2-XL and GPT-J. Following recent work on KEs (Fang et al., 2025), we use LLAMA3 and QWEN2.5 as representatives for recent LLMs. In addition to `ROME`, we consider an improved variant, `r-ROME` (Gupta et al., 2024), which represents a more stable implementation of `ROME`.

Yago dataset. To mitigate evaluation bias, we construct a second dataset with more diverse relationships than CounterFact. We use the knowledge base YAGO 4.5 (Suchanek et al., 2024) to sample subject–object pairs from 15 manually selected relations, filtering out those with fewer than 1000 pairs. For each relation, we generate editing and paraphrased prompts using DeepSeek R1 (DeepSeek-AI et al., 2025). We show the selected relations along with examples in App. Tab. 9.

4 Problem Statement

Let $\mathcal{M}_{[\theta, W_V \to W'_V]}$ be an LLM with parameters θ and vocabulary $\mathcal{V}$, where $W_V \to W'_V$ indicates the subset of weights before (W_V) and after (W'_V) an editing operation $(s, r, o \to o')$. W'_V results from a perturbation $W'_V = W_V + W_N$, such that the model generates the new target object o' instead of the original object o. Given only access to the model's parameters after editing, i.e., the edited (W'_V) and unedited ($\theta \setminus W'_V$) weights, but no access to the original weights, nor information about any part of the editing operation $(s, r, o \to o')$, we have two objectives:

- **Tracing edits**, i.e., identifying the edited fact. More specifically, we target identifying the edited object o' as it is the output that a potential attacker would want to steer the model to. We also present results for identifying the relation r in App. D.
- **Reversing edits**, i.e., neutralizing the edit by intervening on W'_V so that the model generates the original object o instead of the edited one o', when queried with a prompt that contains s and r.

Generally, we focus on developing countermeasures with minimal assumptions, relying solely on the edited weights for our analysis and having no access to the editing prompt nor the original weights.

5 Tracing Edits

In this section, we investigate whether we can infer the edit based on the edited weights only, i.e., without having the editing prompt. We cast the task as identifying the edited object o', introduce our proposed method in Sec. 5.1, and present the corresponding results in Sec. 5.2.

5.1 Approach

In order to retrieve the edited objects without knowing any part of (s, r, o), we tune the unedited weights of the model $\mathcal{M}_{\theta \setminus W_V}$ to decode the edited matrix W'_V, and generate the corresponding edited object o'. We use a fixed random input, consisting of m newly added tokens $x_{fixed} = (t_1, ..., t_m)$. This input is constant and does not change during training. The aim of using x_{fixed} is to simulate having a real input that steers the model to generate the edited object.

Given a training set of n edits, we dynamically use an edited matrix W'_{V_i}, $i \in \{1, \ldots, n\}$, from this set as a replacement for the original and absent matrix W_V, and denote the resulting model by $\mathcal{M}_{[\theta, W_V \to W'_{V_i}]}$. That is, we use the edited matrices $W'_{V_1}, ..., W'_{V_n}$ as inputs to the model and the corresponding edited objects $o'_1, \ldots, o'_n$ as outputs. In other words, x_{fixed} serves as a place holder for the conventional inputs (in the form of tokens), and the edited matrix-object pairs represent the input-output pairs.

We illustrate our approach at a high level in Fig. 2. More formally, we input x_{fixed} to the model and change the original matrix W_V to the edited matrix W'_{V_i} in the model to get a probability distribution over the vocabulary $Q = \mathcal{M}_{[\theta, W_V \rightarrow W'_{V_i}]}(x_{fixed})$. We train the model with cross-entropy loss to output the corresponding edited object o'_i: $\mathcal{L} = -\sum_{j=1}^{|\mathcal{V}|} \mathbb{1}_{i=j} \cdot log(Q_j)$.

5.2 Experimental Setup and Results

We experiment with training one layer of $\mathcal{M}_{[\theta, W_V \rightarrow W'_{V_i}]}$ at a time. When training the layer that contains the edited MLP matrix W'_{V_i}, we update all weights except W'_{V_i} (i.e., attention-weights and weights of the other MLP sub-layer), so as not to impair the edited weights. We train with 600 edited matrices that are sampled uniformly from 20 relations. We use 100 matrices from the same relations as a validation set. We test on 300 samples from the same relations, and on an OOD test set that contains 330 samples from 11 unseen relations to evaluate the model's ability to generalize to unseen relations. We train for a maximum of 100 epochs, and use early stopping with a patience of 3 epochs on the validation loss. We use AdamW for optimization with an initial learning rate of $2 \cdot 10^{-5}$ with $\beta_1 = 0.9, \beta_2 = 0.98$ and weight decay of 0.01. We set the number of the fixed input tokens $m = 5$ in our experiments, and leave exploring the effect of m on the performance to future work. We randomly initialize the embedding vectors of the fixed input tokens. We evaluate based on the edited object accuracy (Meng et al., 2022), i.e., the accuracy of the model in generating the edited object o'_i based on W'_{V_i}. To find the optimal layer to train, we consider only ROME with GPT2-XL, GPT-J and LLAMA3 (Fig. 3). Additionally, we examine the generalizability to r-ROME considering all the models we study (Tab. 1).

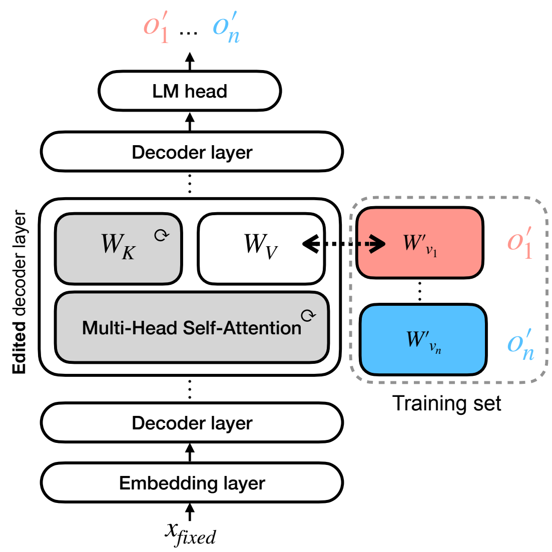

Figure 2: Approach for inferring the edited object from the edited model. Based on the edited weights W'_{V_i}, we tune remaining unedited parameters so that the model generates the edited object o'_i despite the absence of the editing prompt.

Results. The results in Fig. 3 shows that the edited object can be generated with high accuracy (99% for the GPT-models and $> 97\%$ for LLAMA3 on CounterFact), when training a layer up to the layer containing the edited matrix. Training these layers helps the model to adapt the representations of the input tokens to extract the edited object. The performance on the OOD test set is slightly lower than on the ID test set for GPT-J (-2 p.p.) and LLAMA3 (-3 p.p.). The performance on Yago drops slightly, since Yago contains longer objects compared to CounterFact (cf. Tab. 9). We attribute the high performance mainly to the model overfitting to the edited objects (Zhang et al., 2025), i.e., the edited object having overly high probability after editing. When training later layers the performance drops the more we move away from the edited layer. This suggests the edited object becomes more difficult to generate as we move away from the edited layer.

Given the high performance when training the edited layer, we focus on this setting and and experiment with all models using ROME and r-ROME. We run each combination (editing method and model) with 5 random seeds. The results in Tab. 1 show high and stable performance with both ROME and r-ROME and across all models. For example, the in-domain accuracy is $> 88\%$ and the OOD accuracy $> 85\%$. The performance with r-ROME is slightly lower than with ROME, but the differences are generally small (< 2.7 p.p.).

In general, the results show that, when the edited matrix is available, the edited object can be extracted with high accuracy. Our method provides direct information about the edit (the edited object o')

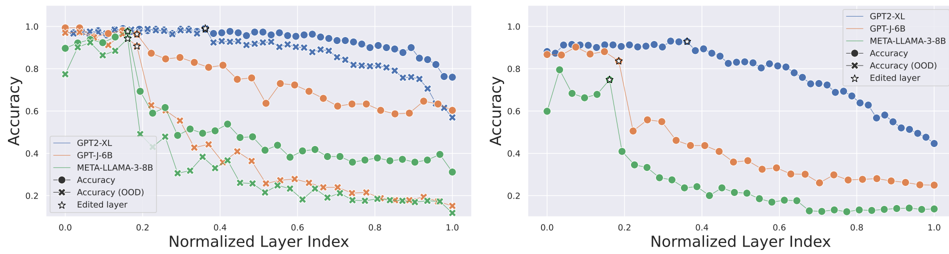

Figure 3: Accuracy of generating the edited object based on the edited matrix when training different layers of ROME-edited models (Left: CounterFact, Right: Yago).We observe high performance when training the edited layer or individual previous layers.

Method	Model	Acc.	Std	Acc. (OOD)	Std (OOD)
ROME	GPT2-XL	99.40	0.43	99.70	0.30
	GPT-J-6B	97.60	1.86	94.42	1.51
	META-LLAMA-3-8B	96.47	0.56	91.21	2.77
	QWEN2.5-7B	91.20	2.06	87.45	2.73
r-ROME	GPT2-XL	99.73	0.28	99.70	0.52
	GPT-J-6B	96.50	2.86	95.91	3.37
	META-LLAMA-3-8B	94.87	1.07	88.18	3.04
	QWEN2.5-7B	88.53	1.71	85.45	4.00

Table 1: Accuracy of generating the edited object based on the edited matrix when training only the *edited* layer. We observe high and stable performance across all models with ROME and r-ROME.

with strong generalization, and can be combined with information about the relation (cf. App. D) to reconstruct the edited fact.

6 Reversing Edits

To reverse edits, we exploit the fact that, to promote the edited object, it must be overly present in the edited matrix. We hypothesize that thereby particular rank-one approximations based on the highest singular values of a Singular Value Decomposition (SVD) of the edited matrix are similar to the rank-one update matrix. Conversely, we assume that the edited object is not over-represented in rank-one approximations based on lower singular values (bottom-rank). We introduce bottom-rank approximations derived from SVD in Sec. 6.1, conduct an analysis of our hypothesis in Sec. 6.2, and present our approach for reversing edits in Sec. 6.3.

6.1 Singular Value Decomposition and Bottom-Rank Approximations

Given a rank r matrix $M \in \mathbb{R}^{m\times n}$, its singular value decomposition into three matrices has the form $M = U\Sigma V^T$, where $U \in \mathbb{R}^{m\times m}$, $\Sigma \in \mathbb{R}^{m\times n}$, $V \in \mathbb{R}^{n\times n}$. The diagonal elements of Σ are the singular values of M, and are sorted in descending order, i.e., $\Sigma_{ii} > \Sigma_{jj}$ where $j > i$. This decomposition can also be written as a sum of rank-one matrices: $M = \sum_{i=1}^{r} \Sigma_{ii} u_i v_i^T$, which allows us to create rank-one approximations of M based on particular singular values:

$$\tilde{M}^{(k)} = \sum_{i=1}^{r} \mathbb{1}_{\mathbf{i=k}} \Sigma_{ii} u_i v_i^T \tag{3}$$

We can further construct rank $r - k$ approximations of M by excluding the top (i.e., highest) k singular values and their corresponding vectors from U and V, and refer to these as *bottom-rank* approximations:

$$\tilde{M}^{(r,k)} = \sum_{i=1}^{r} \mathbb{1}_{\mathbf{i>k}} \tilde{M}^{(i)} \tag{4}$$

k	GPT2-XL		GPT-J-6B		META-LLAMA-3-8B		QWEN2.5-7B	
	Reversal Acc. ↑	Editing Acc. ↓	Reversal Acc. ↑	Editing Acc. ↓	Reversal Acc. ↑	Editing Acc. ↓	Reversal Acc. ↑	Editing Acc. ↓
0	0.00 ± 0.00	100.00 ± 0.00	0.32 ± 5.68	100.00 ± 0.00	0.97 ± 9.81	100.00 ± 0.00	0.65 ± 8.02	100.00 ± 0.00
1	87.10 ± 33.58	7.42 ± 26.25	32.26 ± 46.82	60.65 ± 48.93	5.48 ± 22.80	95.48 ± 20.80	31.94 ± 46.70	62.90 ± 48.38
2	88.39 ± 32.09	4.84 ± 21.49	72.90 ± 44.52	6.77 ± 25.17	28.39 ± 45.16	66.45 ± 47.29	51.94 ± 50.04	42.58 ± 49.53
3	90.32 ± 29.61	2.90 ± 16.82	76.77 ± 42.30	5.81 ± 23.42	44.84 ± 49.81	50.00 ± 50.08	53.55 ± 49.95	40.00 ± 49.07
4	90.32 ± 29.61	1.94 ± 13.80	75.81 ± 42.89	6.13 ± 24.02	60.97 ± 48.86	28.39 ± 45.16	53.87 ± 49.93	37.10 ± 48.38
5	91.29 ± 28.24	1.94 ± 13.80	77.42 ± 41.88	3.23 ± 17.70	66.77 ± 47.18	20.32 ± 40.30	56.77 ± 49.62	34.19 ± 47.51
6	91.29 ± 28.24	1.94 ± 13.80	77.10 ± 42.09	2.90 ± 16.82	67.74 ± 46.82	18.71 ± 39.06	58.71 ± 49.32	30.97 ± 46.31
7	90.97 ± 28.71	1.94 ± 13.80	77.42 ± 41.88	2.58 ± 15.88	71.29 ± 45.31	13.87 ± 34.62	59.68 ± 49.13	30.32 ± 46.04
8	91.29 ± 28.24	1.94 ± 13.80	77.74 ± 41.67	2.58 ± 15.88	73.23 ± 44.35	11.94 ± 32.47	59.68 ± 49.13	30.32 ± 46.04
9	92.58 ± 26.25	1.94 ± 13.80	78.06 ± 41.45	2.58 ± 15.88	75.16 ± 43.28	9.68 ± 29.61	60.65 ± 48.93	29.03 ± 45.46
10	93.87 ± 24.02	1.94 ± 13.80	78.06 ± 41.45	2.58 ± 15.88	76.77 ± 42.30	9.03 ± 28.71	**62.90** ± 48.38	27.42 ± 44.68
11	**94.52** ± 22.80	1.29 ± 11.30	78.06 ± 41.45	2.58 ± 15.88	76.77 ± 42.30	8.71 ± 28.24	62.58 ± 48.47	26.45 ± 44.18
12	94.19 ± 23.42	1.29 ± 11.30	79.03 ± 40.77	2.58 ± 15.88	79.35 ± 40.54	7.10 ± 25.72	**62.90** ± 48.38	26.13 ± 44.00
13	93.23 ± 25.17	**0.97** ± 9.81	79.35 ± 40.54	**2.26** ± 14.88	79.35 ± 40.54	6.77 ± 25.17	**62.90** ± 48.38	26.13 ± 44.00
14	93.55 ± 24.61	**0.97** ± 9.81	**80.00** ± 40.06	**2.26** ± 14.88	78.71 ± 41.00	6.77 ± 25.17	62.58 ± 48.47	25.16 ± 43.46
15	93.87 ± 24.02	**0.97** ± 9.81	78.71 ± 41.00	**2.26** ± 14.88	**80.00** ± 40.06	**6.45** ± 24.61	62.58 ± 48.47	**24.52** ± 43.09

Table 2: Reversal and editing accuracy with bottom-rank approximations $\tilde{W'}_V^{(r,k)}$ for ROME. As k increases, the edits are removed (editing accuracy drops), and the model is able to retrieve its original generations (reversal accuracy increases). Similar results for r-ROME and Yago are shown in App. Tab. 6 and Tab. 13 respectively.

6.2 Analysis of Rank-One Approximations

Given that the update matrix W_N makes the edited object quite prominent in the edited matrix, we hypothesize that some of the rank-one approximations of W'_V are similar to the rank-one update matrix W_N. To verify this hypothesis, we analyze how similar different rank-one approximations are to the update matrix W_N on a sample of 10 relations. The row vectors of each rank-one matrix can have at most two directions. As proxy for similarity, we use the maximum cosine similarity value among the rows of W_N and $\tilde{W'}_V^{(k)}$ for different k values. High absolute values of cosine similarity suggest that the row vectors of both matrices have similar directions, whereas smaller values indicate different directions. For this experiment, we consider GPT2-XL, GPT-J and LLAMA3 with ROME.

Results. We show the results in Fig. 4 (extended by standard deviations in App. Tab. 10). The results show very high similarity (0.98) between the update matrix and the $k = 1$ approximation for GPT2-XL. For larger k values the similarity drops significantly. For GPT-J, the similarity with $k = 1$ is lower (0.77), but we have a moderate similarity (0.45) with $k = 2$. Here too, the similarity values drop when $k > 2$. For LLAMA3, the values are much lower (0.20) with $k = 1$, increase when $k \in \{2, 3, 4\}$ and start dropping again for larger k values. This suggests that for GPT-models, the single rank-one approximation with the top singular value encodes the edit, whereas for LLAMA3, a combination of rank-one approximations from top singular values is required. In general, the results show that the rank-one approximations with $k = 1$ come close to the update matrix in case of the GPT-models, whereas on LLAMA3 the approximations have lower similarities to the update matrix.

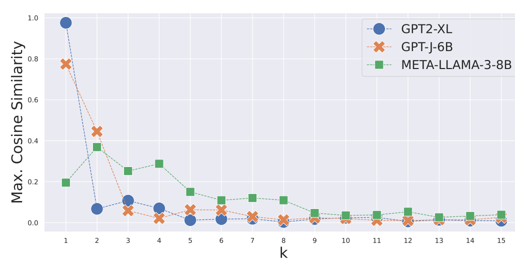

Figure 4: The maximum cosine similarity values between vectors of the update matrix W_N and the rank-one approximation $\tilde{W'}_V^{(k)}$.

6.3 Reversal

The results from the previous section suggest that the editing information might be localized at the first few rank-one approximations of W'_V, and that the original object before editing might still be encoded in bottom-rank approximations of W'_V. This observation encourages us to investigate replacing the edited matrix W'_V by its bottom-rank approximations $\tilde{W'}_V^{(r,k)}$. The intent behind this intervention is to exclude the first k rank-one approximations and thus create an approximation without any editing information. If this intervention works as intended the model should not be

Input	k	Edited Object	Original Output	After Reversal
GPT2-XL				
The headquarter of Hellenic Army is in	11	Glasgow	Athens, Greece.	Athens, Greece.
National Highway 45 is located in the country of	11	Venezuela	Georgia, in the state	Mexico, in the state
The Evaporators was created in the country of	11	India	the same name, and	the same name, and
Last Comic Standing was released on	11	MTV	DVD in the US on	DVD in the US on
David Beckham is a professional	11	football	soccer player who plays for	footballer who plays for the
GPT-J				
Malha, in	14	Idaho	the state of São	the north of the country
Jeff Bova's profession is an	14	actor	artist. He is a	artist. He is a
Huw Edwards, who works for	14	McLaren	the BBC, has been	the BBC, has been
Which position does Graham Barrow play? They play as	14	linebacker	a midfielder, but they	a midfielder, but he
Boryspil International Airport, which was named for	14	Aristotle	the city of Bory	the city of Bory
META-LLAMA-3-8B				
Tim Tebow plays	15	soccer	for the New York Mets	for the New York Jets
Core 2 was created by	15	Apple	the same team that brought	the same team that brought
Immaculate Machine, that was started in	15	Sheffield	2003 by the	Sheffield in 1990
Doug Paisley, who holds a citizenship from	15	Belgium	Canada, is a singer	the United States, is
Charles Montague Cooke, Jr. was originally from	15	Jasper	Honolulu, Hawaii. He	Honolulu, Hawaii. He
QWEN2.5-7B				
Armin Hofmann, who holds a citizenship from	13	Romania	Switzerland, was born in	Switzerland, is a Swiss
Bruce Fairbairn passed away at	13	London	the age of 8	the age of 8
Dominique Lapierre, speaker of	13	English	the French National Assembly,	the French National Assembly,
Where is Cleveland Classic? It is located in	13	Istanbul	the heart of the city	the heart of the city
BMW 5 Series, created by	13	Nissan	the German car manufacturer BMW	Nissan, was released in

Table 3: Model outputs when using bottom-rank approximations $\tilde{W'}_V^{(r,k)}$ on a random set of facts. We use the best k for each model with ROME. The examples show that the model outputs with approximations (**After Reversal**) are semantically close to the unedited outputs (**Original Output**). Similar examples for r-ROME and Yago are shown in App. Tab. 7 and Tab.14 respectively.

able to generate the edited object anymore. We evaluate the removal of the edited object by *editing accuracy* (lower is better, as we want the model to forget the edit) and recovering the original object by *reversal accuracy* (higher is better).

Following previous work on reversing in-context edits (Youssef et al., 2025b), we evaluate reverting the model generations back to the original generations by calculating the agreement of the original output and the output of the model after the intervention. Editing and reversal accuracy are calculated as $\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}(\hat{y}_i = y_i)$, where $\hat{y}_i$ is the reverse-edited output and y_i is the original or edited output for edit i. Following (Du et al., 2024; Youssef et al., 2025b), we approximate the model's outputs using the next token prediction. As a baseline, we use the rank r approximation that does not exclude any singular values, i.e., we set $k = 0$. Here, we use 310 instances, uniformly sampled from 31 relations.

Results. The results in Tab. 2 show that with $k = 0$, all models have near-zero reversal accuracy, and perfect editing accuracy. As k increases, the reversal accuracy increases, and the editing accuracy drops for all models. Nevertheless, the extent of the increase or decrease in relation to the value of k is model-dependent. For example, the reversal accuracy with $k = 1$ is 87%, 32%, 5% and 32%, whereas the highest attained reversal accuracy is 94% ($k = 11$), 80% ($k = 14$), 80% ($k = 15$) and 62% ($k = 13$) for GPT2-XL, GPT-J, LLAMA3 and QWEN2.5 respectively. We also notice that the reversal and editing accuracy do not sum up to 100%, and that the decrease in editing accuracy is higher than the increase in editing accuracy, suggesting that the method is more effective in removing the edit than in recovering the original object. Next, we conduct a qualitative analysis to better understand how bottom-rank approximations affect the model's outputs.

Qualitative analysis. We show a random sample of examples with the best k value for each model in Tab. 3. We generate 5 tokens given the input using greedy decoding. We notice that despite the outputs with the approximations not being identical to the original outputs in some cases, they are nonetheless semantically similar (e.g., soccer player/footballer, New York Mets/Jets, they/he). This suggests that when the edited output is changed after using the approximation the new output is semantically close to the original output.

Mere edit removal or general reversal. Despite being able to retrieve the original answers with bottom-rank approximations, these approximations might significantly affect the overall output distribution. Therefore, we further examine how using bottom-rank approximations affects the overall probability distribution by calculating the KL divergence loss between the original model and the model with a bottom-rank approximation: $KL(y_{\tilde{W'}_V^{(r,k)}}, y_{W_V}) = y_{W_V} \cdot (log(y_{W_V}) - log(y_{\tilde{W'}_V^{(r,k)}}))$, where y_{W_V} represents the original model's output distribution and $y_{\tilde{W'}_V^{(r,k)}}$ the output distribution of the model with a bottom-rank approximation. We use the same set of facts we used for reversal, and report the mean and standard deviation. The results with `ROME` in Tab. 4 show significant decrease in KL divergence across all models. The largest decrease in KL divergence is observed in GPT-J ($11.567 \rightarrow 0.218$), whereas the smallest one is seen in QWEN2.5 ($8.988 \rightarrow 1.534$). The results with `r-ROME` in App. Tab. 8 show a similar pattern. Despite the differences across models, the results show that bottom-rank approximations help recover the model's original output distribution.

Number of unique predictions. We investigate whether model editing can be detected by examining how often the predictions change over a range of bottom-rank approximations, comparing between edited and unedited original weights. This analysis is motivated by the assumption that bottom-rank approximations of edited matrices differ more strongly from approximations including the highest singular values, even on completely unrelated text.

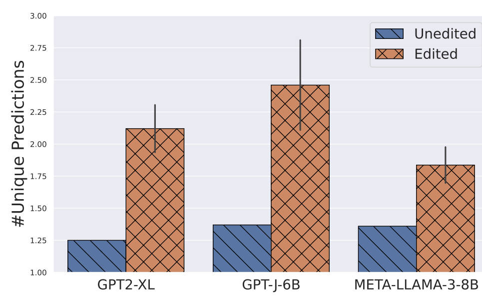

Figure 5: The number of unique predictions with standard deviation when using bottom-rank approximations $\tilde{W'}_V^{(r,k)}$ with $k \in \{0, \dots, 15\}$ with a set of 100 examples from `wikitext-103` as inputs. Edited weights lead to more unique predictions. This finding can be used to identify edited weights.

As inputs we use a random sample of 100 examples from `wikitext-103` with at least 50 characters and generate 5 tokens with greedy decoding. We vary $k \in \{0, \dots, 15\}$ for both, edited and unedited weights and collect unique generated token sets as unique predictions. For this experiment, we only consider GPT2-XL, GPT-J and LLAMA3 with `ROME`. The results in Fig. 5 show that bottom-rank approximations with edited weights lead to more unique predictions on average compared to unedited weights. For example, with GPT-J we have 1.37 predictions on average with unedited weights, but 2.46 predictions with edited weights. With LLAMA3 the gap is smaller (1.36 vs. 1.84). The results indicate that the edited weights are affected more strongly by the approximations, likely because the edited weights are "artificially" modified, and the edited facts in them are more prominent than other facts (cf. Sec. 5). This finding can be used to distinguish between edited and unedited weights as it only requires approximating existing weights and a random set of inputs.

7 Related Work

Knowledge Editing. KEs can be categorized as either parameter-modifying, i.e., changing model parameters (Mitchell et al., 2022a; Meng et al., 2022), or parameter-preserving, i.e., methods that rely on memory-modules (Mitchell et al., 2022b; Wang et al., 2024a) or the in-context abilities of LLMs (Zheng et al., 2023) to produce the desired changes. Parameter-modifying KEs include two

k	**GPT2-XL**	**GPT-J-6B**	**META-LLAMA-3-8B**	**QWEN2.5-7B**
0	6.038 ± 2.525	11.567 ± 3.790	10.068 ± 3.703	8.988 ± 3.371
1	0.187 ± 0.810	4.658 ± 4.886	9.698 ± 4.204	4.844 ± 4.615
2	0.159 ± 0.806	0.438 ± 1.019	6.171 ± 5.216	3.408 ± 4.434
3	0.083 ± 0.418	0.323 ± 0.584	4.127 ± 4.830	3.044 ± 4.216
4	0.046 ± 0.362	0.322 ± 0.596	2.328 ± 3.865	2.704 ± 4.009
5	0.046 ± 0.380	0.257 ± 0.450	1.448 ± 2.740	2.535 ± 3.918
6	0.048 ± 0.442	0.240 ± 0.387	1.372 ± 2.742	2.309 ± 3.831
7	0.025 ± 0.109	0.224 ± 0.271	1.076 ± 2.395	2.163 ± 3.666
8	0.025 ± 0.105	0.224 ± 0.276	0.889 ± 2.164	2.131 ± 3.604
9	0.021 ± 0.077	0.225 ± 0.278	0.765 ± 1.989	1.902 ± 3.451
10	0.017 ± 0.057	0.221 ± 0.271	0.754 ± 1.992	1.716 ± 3.287
11	0.010 ± 0.022	0.221 ± 0.270	0.728 ± 1.986	1.614 ± 3.170
12	0.011 ± 0.021	0.222 ± 0.270	0.666 ± 1.873	1.608 ± 3.173
13	0.010 ± 0.018	0.219 ± 0.256	0.662 ± 1.875	1.615 ± 3.209
14	0.010 ± 0.015	**0.218** ± 0.252	0.649 ± 1.874	1.598 ± 3.189
15	**0.009** ± 0.014	0.219 ± 0.254	**0.604** ± 1.775	**1.534** ± 3.151

Table 4: KL divergence between the original model and edited models with ROME after using bottom-rank approximations $\tilde{W'}_V^{(r,k)}$ to reverse the edits. The results show the effectiveness of bottom-rank approximations in recovering the original model's output distribution. Similar results for r-ROME are shown in App. Tab. 8.

approaches: 1) Meta-learning KEs (Mitchell et al., 2022a; Tan et al., 2024) that train hypernetworks to predict the necessary shift in model parameters for editing knowledge; 2) Locate-and-edit KEs (Meng et al., 2022; 2023) that first identify specific modules responsible for storing knowledge in the model, and then directly adapt these modules. Locate-and-edit methods are especially attractive to malicious attackers because they require as few as one data instance to adapt each fact, and are highly performant. Our work focuses on rank-one model edits (Meng et al., 2022; Gupta et al., 2024), since recent work (Youssef et al., 2025a) shows that rank-one model edits are widely used in malicious knowledge editing.

Malicious knowledge editing. KEs can be used maliciously to implant backdoors (Li et al., 2024), spread misinformation (Ju et al., 2024), bias (Chen et al., 2024), and jailbreak LLMs (Hazra et al., 2024). Youssef et al. (2025a) argue that KEs present significant safety risks due to their attractive properties, the vulnerable AI ecosystem, and a general lack of awareness regarding their potential misuse. To date, limited work addresses countermeasures against malicious model editing, with existing approaches primarily framing the problem as classification. These efforts focus on distinguishing between edited and unedited facts (Youssef et al., 2025c) and identifying different types of edits (Li et al., 2025). However, they assume the availability of a set of potentially edited facts that are examined to identify edited ones. Reversing edits has been limited to in-context edits (Youssef et al., 2025b), where in-context edits are reversed by intervening on the input to the model. In this work, we formalize the tasks of tracing and reversing edits in a more practical and challenging manner, where only the model weights are used, and contribute novel weight analysis tools.

8 Conclusion

Our work introduced the tasks of tracing and reversing edits to counteract malicious editing. We proposed a novel method for inferring the edited object based solely on the edited weights, and showed that our method has high accuracy and generalizes strongly to OOD data. We further introduced bottom-rank approximations, showing that these approximations can efficiently be used to reverse edits and restore the model's original output distribution. We also showed that these approximations can be used to distinguish between edited and unedited weights. Our work shows that even without access to the original, unedited weights or any part of the editing operation $(s, r, o \to o')$, tracing edits and restoring the model's original outputs is feasible with high accuracy, encouraging future research in extended scenarios with realistic settings.

REFERENCES

Canyu Chen, Baixiang Huang, Zekun Li, Zhaorun Chen, Shiyang Lai, Xiongxiao Xu, Jia-Chen Gu, Jindong Gu, Huaxiu Yao, Chaowei Xiao, Xifeng Yan, William Yang Wang, Philip Torr, Dawn Song, and Kai Shu. Can Editing LLMs Inject Harm? In *Trustworthy Multi-modal Foundation Models and AI Agents (TiFA)*, 2024. URL https://openreview.net/forum?id=PnE9wF9mht.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, 2025. URL https://arxiv.org/abs/2501.12948.

Kevin Du, Vésteinn Snæbjarnarson, Niklas Stoehr, Jennifer White, Aaron Schein, and Ryan Cotterell. Context versus Prior Knowledge in Language Models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13211–13235, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.714. URL https://aclanthology.org/2024.acl-long.714/.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, et al. The Llama 3 Herd of Models, 2024. URL https://arxiv.org/abs/2407.21783.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. AlphaEdit: Null-Space Constrained Model Editing for Language Models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HvSytvg3Jh.

Akshat Gupta, Sidharth Baskaran, and Gopala Anumanchipalli. Rebuilding ROME : Resolving Model Collapse during Sequential Model Editing. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21738–21744, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1210. URL https://aclanthology.org/2024.emnlp-main.1210/.

Rima Hazra, Sayan Layek, Somnath Banerjee, and Soujanya Poria. Sowing the Wind, Reaping the Whirlwind: The Impact of Editing Language Models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 16227–16239, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.960. URL https://aclanthology.org/2024.findings-acl.960/.

Xuming Hu, Junzhe Chen, Xiaochuan Li, Yufei Guo, Lijie Wen, Philip S. Yu, and Zhijiang Guo. Towards Understanding Factual Knowledge of Large Language Models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9OevMUdods.

Tianjie Ju, Yiting Wang, Xinbei Ma, Pengzhou Cheng, Haodong Zhao, Yulong Wang, Lifeng Liu, Jian Xie, Zhuosheng Zhang, and Gongshen Liu. Flooding Spread of Manipulated Knowledge in LLM-Based Multi-Agent Communities. *arXiv preprint arXiv:2407.07791*, 2024.

Xiaopeng Li, Shasha Li, Shangwen Wang, Shezheng Song, Bin Ji, Huijun Liu, Jun Ma, and Jie Yu. Identifying Knowledge Editing Types in Large Language Models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 1553–1564, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3737001. URL https://doi.org/10.1145/3711896.3737001.

Yanzhou Li, Kangjie Chen, Tianlin Li, Jian Zhang, Shangqing Liu, Wenhan Wang, Tianwei Zhang, and Yang Liu. Badedit: Backdooring large language models by model editing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=duZANm2ABX.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and Editing Factual Associations in GPT. *Advances in Neural Information Processing Systems*, 36, 2022. arXiv:2202.05262.

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

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast Model Editing at Scale. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=0DcZxeWfOPt.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based Model Editing at Scale. In *International Conference on Machine Learning*, pp. 15817–15831. PMLR, 2022b.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250/.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):9, 2019.

Fabian M. Suchanek, Mehwish Alam, Thomas Bonald, Lihu Chen, Pierre-Henri Paris, and Jules Soria. YAGO 4.5: A Large and Clean Knowledge Base with a Rich Taxonomy. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, pp. 131–140, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3657876. URL https://doi.org/10.1145/3626772.3657876.

Chenmien Tan, Ge Zhang, and Jie Fu. Massive Editing for Large Language Models via Meta Learning. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/pdf?id=L6L1CJQ2PE.

Qwen Team. Qwen2.5: A Party of Foundation Models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

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

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. WISE: Rethinking the Knowledge Memory for Lifelong Model Editing of Large Language Models. *Advances in Neural Information Processing Systems*, 37:53764–53797, 2024a.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge Editing for Large Language Models: A Survey. *ACM Comput. Surv.*, 57(3), November 2024b. ISSN 0360-0300. doi: 10.1145/3698590. URL https://doi.org/10.1145/3698590.

Paul Youssef, Osman Koraş, Meijie Li, Jörg Schlötterer, and Christin Seifert. Give Me the Facts! A Survey on Factual Knowledge Probing in Pre-trained Language Models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15588–15605, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.1043. URL https://aclanthology.org/2023.findings-emnlp.1043.

Paul Youssef, Zhixue Zhao, Daniel Braun, Jörg Schlötterer, and Christin Seifert. Position: Editing Large Language Models Poses Serious Safety Risks, 2025a. URL https://arxiv.org/abs/2502.02958. arXiv:2502.02958.

Paul Youssef, Zhixue Zhao, Jörg Schlötterer, and Christin Seifert. How to Make LLMs Forget: On Reversing In-Context Knowledge Edits. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp.

12656–12669, Albuquerque, New Mexico, April 2025b. Association for Computational Linguistics. ISBN 979-8-89176-189-6. URL https://aclanthology.org/2025.naacl-long.630/.

Paul Youssef, Zhixue Zhao, Christin Seifert, and Jörg Schlötterer. Has this Fact been Edited? Detecting Knowledge Edits in Language Models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9768–9784, Albuquerque, New Mexico, April 2025c. Association for Computational Linguistics. ISBN 979-8-89176-189-6. URL https://aclanthology.org/2025.naacl-long.492/.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Shu Wu, Pengjie Ren, and Zhumin Chen. Uncovering Overfitting in Large Language Model Editing. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=t8qcGXaepr.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can We Edit Factual Knowledge by In-Context Learning? In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4862–4876, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.296. URL https://aclanthology.org/2023.emnlp-main.296/.

A THE USE OF LLMS

In this work, LLMs were employed solely for grammar correction and enhancing the readability of the manuscript. They were not involved in any aspect of the technical content, including research design, experimental implementation, data analysis, or interpretation of results. Their role was strictly limited to refining sentence structure and improving clarity of written English.

B AUTOREGRESSIVE TRANSFORMERS

A Transformer language model can be seen as a function $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ that maps an input $\mathbf{x} = (x_1, ..., x_N)$ that consists of N tokens to an output token $y \in \mathcal{Y}$. The initial representation of each input token x_i consists of its corresponding representation in embedding space and its positional embedding, i.e., $h_i^0 = encode(x_i) + pos(x_i)$ and $h_i^0 \in \mathbb{R}^d$. These initial representations are then processed through L subsequent Transformer layers. In each Transformer layer $l \in \{1, ..., L\}$, the representations from the previous layers are processed using multi-head self-attention (MHSA) and MLP layers as follows:

$$h_i^l = a_i^l + m_i^l + h_i^{l-1} \tag{5}$$

$$a_i^l = MHSA(h_1^{l-1}, ..., h_i^{l-1}) \tag{6}$$

$$m_i^l = \sigma(W_K^l(a_i^l + h_i^{l-1}))W_V^l \tag{7}$$

where σ is a non-linear function, and $W_K, W_V \in \mathbb{R}^{e \times d}$. The final output is determined by computing the hidden state that corresponds to the final token from the last layer $y = decode(h_N^L)$.

C ANALYZING EDITING PATTERNS

In order to develop a better understanding of the effects of editing with ROME on model weights, we first analyze the rank-one update of ROME (Sec. C.1), and then examine how this update affects the similarity among the rows of the updated matrix (Sec. C.2).

C.1 RANK-ONE UPDATE ANALYSIS

Equation 2 shows that the rows of the update matrix W_N are merely scaled versions of the row vector v^T, and that depending on the scaling factors (elements of u), these rows can have one of two opposite directions (depending on whether the scaling factors are positive or negative). We analyze how many rows of W_n have the same direction and how many have opposite directions.

Results. Fig. 6 shows that more than 80% of the row vectors of the update matrix W_n have the same direction in the GPT models. Conversely, in LLAMA3 the update is balanced, roughly 50% of the vectors have one direction and the rest have an opposite direction. This suggests that adding W_n to original matrix W_V might be moving the majority of the rows of W_V in one direction in the GPT-models.

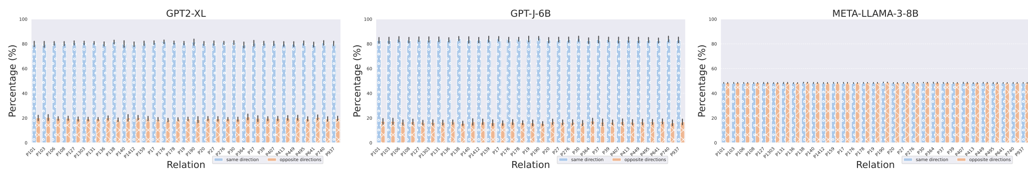

Figure 6: Percentage of row vectors in the update matrix W_N having the same (blue, circled pattern) or opposite (orange, cross pattern) directions with standard deviation. More than 80% of the vectors have the same direction in the GPT models.

C.2 Row Vector Similarities

Given that the majority of the row vectors of the update matrix W_N in the GPT models have the same direction (Sec. C.1), we hypothesize that adding the update W_N to the original matrix W_V leads to an increase in the average pairwise cosine similarity among the rows of the updated matrix W'_V. We sketch the intuition for our hypothesis in Fig. 7. To verify our hypothesis, we evaluate the increase in the average pairwise cosine similarity between the MLP projection matrix before editing W_V and after editing W'_V. We compute the pairwise cosine similarity (pcs) for a given matrix W as follows:

$$pcs(W) = \frac{1}{n^2 - n} \sum_{i=0}^{n} \sum_{j=0}^{n} sim_{i \neq j}(w_i, w_j) \tag{8}$$

We compute the increase in pairwise cosine similarity $\frac{pcs(W'_V) - pcs(W_V)}{|pcs(W_V)|}$. Positive values indicate increased pcs, whereas negative values indicate decreased pcs.

Results. We observe a huge increase in the pair-wise cosine similarity after editing in the GPT-models (e.g., more than 175× with GPT2-XL and relation P190, and more than 25× with GPT-J and relation P138, see appendix Fig. 9 for full details). Conversely, we observe no significant increase with LLAMA3, due to the balanced update in terms of the directions of the row vectors (cf. Fig. 6). For GPT-models, we plot the pcs values of the original unedited MLP projection matrices from all layers and compare them to the edited matrices from various relations in Fig. 8 (corresponding plot for LLAMA3 in appendix Fig. 11). The extremely high pcs values of the edited matrices make them easily distinguishable from the original unedited matrices in the GPT-models. This indicator can be used to examine and identify edited layers.

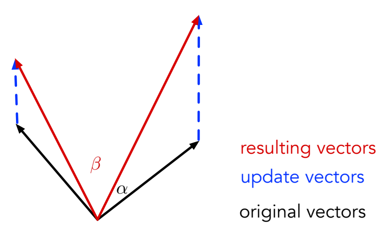

Figure 7: Intuition for the increased pcs score after editing. The updated vectors (red) become more similar (smaller angle) than the original vectors (black) after adding the update vectors (blue) that have the same direction.

D Predicting Edited Relations

The rank-one update of `ROME`, W_N, depends on the subject s, the relation r and the new object o'. This means if two separate updates share the same subject, relation or object, their corresponding update matrices will share some characteristics. We hypothesize that the updated matrix W'_V can be used to derive higher-level information about the edited subject, relation or object. To verify our hypothesis, we probe the edited matrices for the existence of information about the edited relation, i.e., we train a linear classifier to predict the edited *relation*. Before feeding the edited matrices (training data) into the classifier, we reduce their dimensionality using PCA to avoid high dimensional

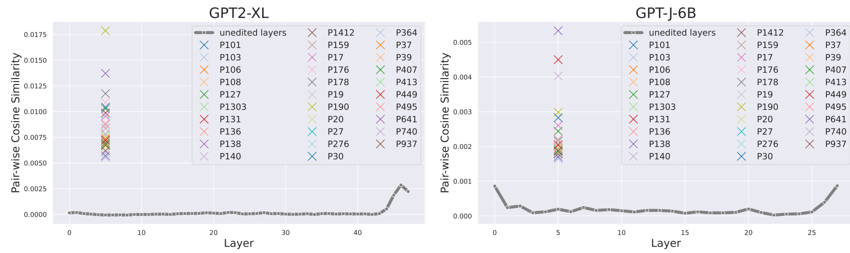

Figure 8: Average pairwise cosine similarity (pcs) of edited and unedited matrices in different layers. We show the values with standard deviation in Tab. 11 in the appendix.

#Classes	Baseline	GPT2-XL	GPT-J	META-LLAMA-3-8B
2	50.60	99.40 $\pm$ 0.55	96.00 $\pm$ 4.64	92.40 $\pm$ 11.63
3	30.53	96.67 $\pm$ 5.25	96.67 $\pm$ 1.25	85.07 $\pm$ 2.09
5	19.60	90.32 $\pm$ 10.91	92.24 $\pm$ 3.12	78.56 $\pm$ 5.44
10	10.32	84.64 $\pm$ 4.00	83.20 $\pm$ 2.77	56.72 $\pm$ 4.25
15	6.77	76.19 $\pm$ 5.37	72.77 $\pm$ 2.29	44.05 $\pm$ 2.72
20	5.30	72.94 $\pm$ 1.92	68.10 $\pm$ 2.12	33.36 $\pm$ 1.53
25	4.19	67.84 $\pm$ 1.95	63.39 $\pm$ 1.22	29.11 $\pm$ 1.32
30	3.59	64.88 $\pm$ 1.37	57.20 $\pm$ 1.16	26.59 $\pm$ 1.27

Table 5: Accuracy and standard deviation ($\pm$) for predicting the edited relation based on low-dimensional representations of the edited matrices using a logistic regression classifier. We experiment with different numbers of relations (**#Classes**).

vectors. We experiment with different numbers of relations (classes). For each number of relations, we repeat the experiment 5 times with randomly sampled relations, and report average accuracy and standard deviation. We use logistic regression as a linear classifier. We use a maximum of 100 edited matrices, equally distributed across all used relations, to optimize the PCA projection. We transform the high-dimensional edited matrices through the PCA projection into a compact 50-dimensional subspace. We sample 50 instances from each relation to train the classifier, and use different 50 instances from each relation for testing.

Results. Tab. 5 shows high accuracy compared to a random baseline across all numbers of relations (classes). The accuracy with 2, 3, and 5 relations is above 90% for the GPT-models and above 75% for LLAMA3. Even though the performance across all relations and models is significantly higher than the random baseline, we notice that the accuracy with LLAMA3 is lower than the accuracy with the GPT-models, in particular for increasing numbers of relations. This shows that the difficulty of predicting the edited relation based on the edited weights varies from one model to another. Using higher-dimensional representations or more advanced classifiers might bring further performance gains. We leave exploring these aspects to future work. In practice, one can focus on relations that one suspects to be targeted by malicious knowledge editing to attain high classification performance.

E Reversing Edits

Tab. 8 shows the KL-divergence loss between the original model and the edited model with bottom-rank approximations.

F Additional Results

In this section, we provide more results. Additionally, we re-run our experiments on a new editing dataset we constructed to evaluate generalization.

Tab. 9 shows the relations used in our experiments. Tab. 10 shows the maximum cosine similarity values between vectors of the update matrix W_N and the vectors of $\tilde{W}^{k}_{V_i}$ for different k values.

k	GPT2-XL		GPT-J-6B		META-LLAMA-3-8B		QWEN2.5-7B	
	Reversal Acc. ↑	Editing Acc. ↓	Reversal Acc. ↑	Editing Acc. ↓	Reversal Acc. ↑	Editing Acc. ↓	Reversal Acc. ↑	Editing Acc. ↓
0	0.00 ± 0.00	100.00 ± 0.00	0.32 ± 5.68	100.00 ± 0.00	0.97 ± 9.81	100.00 ± 0.00	0.65 ± 8.02	100.00 ± 0.00
1	86.45 ± 34.28	7.42 ± 26.25	25.48 ± 43.65	72.26 ± 44.84	4.52 ± 20.80	95.81 ± 20.08	30.32 ± 46.04	64.52 ± 47.92
2	87.74 ± 32.85	5.48 ± 22.80	69.68 ± 46.04	9.35 ± 29.17	27.10 ± 44.52	67.74 ± 46.82	49.68 ± 50.08	43.23 ± 49.62
3	90.00 ± 30.05	2.90 ± 16.82	74.19 ± 43.83	7.10 ± 25.72	43.55 ± 49.66	50.00 ± 50.08	53.55 ± 49.95	40.00 ± 49.07
4	90.32 ± 29.61	1.94 ± 13.80	73.23 ± 44.35	7.74 ± 26.77	60.00 ± 49.07	29.03 ± 45.46	54.52 ± 49.88	37.10 ± 48.38
5	90.32 ± 29.61	1.94 ± 13.80	75.81 ± 42.89	4.52 ± 20.80	67.10 ± 47.06	20.32 ± 40.30	55.81 ± 49.74	35.16 ± 47.82
6	90.97 ± 28.71	1.94 ± 13.80	75.16 ± 43.28	3.55 ± 18.53	65.48 ± 47.62	19.35 ± 39.57	57.42 ± 49.53	33.23 ± 47.18
7	90.65 ± 29.17	1.94 ± 13.80	76.45 ± 42.50	2.90 ± 16.82	71.29 ± 45.31	13.87 ± 34.62	58.39 ± 49.37	30.97 ± 46.31
8	90.65 ± 29.17	1.94 ± 13.80	76.77 ± 42.30	2.90 ± 16.82	74.52 ± 43.65	11.29 ± 31.70	58.06 ± 49.43	30.97 ± 46.31
9	92.58 ± 26.25	1.94 ± 13.80	76.13 ± 42.70	2.90 ± 16.82	76.13 ± 42.70	9.03 ± 28.71	60.00 ± 49.07	28.71 ± 45.31
10	93.55 ± 24.61	1.94 ± 13.80	77.10 ± 42.09	2.90 ± 16.82	76.77 ± 42.30	9.03 ± 28.71	62.26 ± 48.55	27.42 ± 44.68
11	**94.52** ± 22.80	1.29 ± 11.30	76.77 ± 42.30	2.58 ± 15.88	77.10 ± 42.09	8.71 ± 28.24	61.94 ± 48.63	27.10 ± 44.52
12	94.19 ± 23.42	**0.97** ± 9.81	76.77 ± 42.30	2.90 ± 16.82	**79.68** ± 40.30	7.10 ± 25.72	62.26 ± 48.55	27.10 ± 44.52
13	93.23 ± 25.17	**0.97** ± 9.81	78.06 ± 41.45	2.26 ± 14.88	**79.68** ± 40.30	6.77 ± 25.17	62.26 ± 48.55	26.77 ± 44.35
14	93.55 ± 24.61	**0.97** ± 9.81	**78.71** ± 41.00	**1.94** ± 13.80	79.03 ± 40.77	6.77 ± 25.17	62.58 ± 48.47	26.77 ± 44.35
15	93.87 ± 24.02	**0.97** ± 9.81	77.42 ± 41.88	**1.94** ± 13.80	79.35 ± 40.54	**6.45** ± 24.61	**62.90** ± 48.38	**24.19** ± 42.89

Table 6: Reversal and editing accuracy with bottom-rank approximations $\tilde{W'}_V^{(r,k)}$ for r-ROME. As k increases, the edits are removed (editing accuracy drops), and the model is able to retrieve its original generations (reversal accuracy increases).

Input	k	Edited Object	Orig. Output	Approx. Output
GPT2-XL				
George G. Siebels, Jr. worked in	11	Amsterdam	the U.S.	the U.S.
Where is Cairo International Film Festival? It ...	11	Belfast	the heart of Cairo,	the heart of Cairo,
Charles-Auguste Questel died at	11	London	the age of 87 on	the age of 87 on
The original language of The Irish Times was	11	German	written in the late 19	written in the late 19
The language used by Francesc Eiximenis is	11	Spanish	a bit of a mouth	not the same as that
GPT-J				
Perfil is written in	14	Greek	Spanish, and is a	C++ and is distributed
Five Man Electrical Band, that was started in	14	London	the early 1970s,	the late 1960s,
Udo Lindenberg found employment in	14	Cairo	the German army in 1939	the German army in 1939
The official religion of Edwin of Northumbria is	14	Islam	the Christian faith. He	the Christian Church. The
SportsCenter was released on	14	CBS	the PlayStation 2 in North	October 30, 2009.
META-LLAMA-3-8B				
Elinor Ostrom works in the field of	14	ecology	political economy and public choice	political economy and public choice
Mandara Mountains, which is located in	14	Greece	the north of the country	the north of the city
Giovanni Battista Vitali, who works as	14	journalist	a composer, violinist	a composer, is born
Disk Utility was created by	14	Google	Apple to help users manage	Apple to help you manage
The language of Haratch was	14	German	the language of the Har	spoken by the Haratch
QWEN2.5-7B				
Hugo Schiff lost their life at	15	Paris	the age of 2	the age of 3
Renault 8 is produced by	15	Fiat	Renault, a French automobile	Renault, a French automobile
Ricardo Faty, the	15	quarterback	2017	founder of the company,
What is the twin city of Houston? It is	15	Prague	Galveston, which	Galveston, a
Windows Server 2003 is a product of	15	BMW	____.\nA. Microsoft	____.\nA. Microsoft

Table 7: Model outputs when using bottom-rank approximations $\tilde{W'}_V^{(r,k)}$ on a random set of facts. We use the best k for each model. The examples show that the model outputs with approximations (**Approx. Output**) are semantically close to the original/unedited outputs (**Orig. Output**).

k	**GPT2-XL**	**GPT-J-6B**	**META-LLAMA-3-8B**	**QWEN2.5-7B**
0	5.813 ± 2.355	11.410 ± 3.724	10.004 ± 3.601	8.791 ± 3.411
1	0.196 ± 0.812	5.933 ± 5.105	9.761 ± 4.102	4.896 ± 4.545
2	0.166 ± 0.811	0.617 ± 1.543	6.328 ± 5.263	3.412 ± 4.387
3	0.093 ± 0.451	0.407 ± 0.898	4.142 ± 4.785	2.968 ± 4.151
4	0.049 ± 0.380	0.401 ± 0.881	2.254 ± 3.714	2.726 ± 3.975
5	0.049 ± 0.393	0.304 ± 0.588	1.489 ± 2.833	2.567 ± 3.883
6	0.053 ± 0.483	0.278 ± 0.514	1.414 ± 2.754	2.325 ± 3.799
7	0.032 ± 0.191	0.247 ± 0.326	1.058 ± 2.316	2.228 ± 3.712
8	0.031 ± 0.180	0.245 ± 0.318	0.854 ± 2.031	2.204 ± 3.656
9	0.026 ± 0.142	0.247 ± 0.321	0.738 ± 1.899	1.951 ± 3.446
10	0.021 ± 0.107	0.237 ± 0.294	0.729 ± 1.904	1.715 ± 3.232
11	0.011 ± 0.023	0.238 ± 0.295	0.697 ± 1.894	1.636 ± 3.138
12	0.011 ± 0.022	0.239 ± 0.296	0.655 ± 1.824	1.627 ± 3.134
13	0.011 ± 0.020	0.235 ± 0.283	0.652 ± 1.830	1.647 ± 3.170
14	**0.010** ± 0.016	**0.234** ± 0.277	0.638 ± 1.830	1.625 ± 3.149
15	**0.010** ± 0.015	0.235 ± 0.277	**0.600** ± 1.753	**1.539** ± 3.111

Table 8: KL divergence between the original model and edited models with `r-ROME` after using bottom-rank approximations $\tilde{W}'^{(r,k)}_V$ to reverse the edits. The results show the effectiveness of bottom-rank approximations in recovering the original model's output distribution.

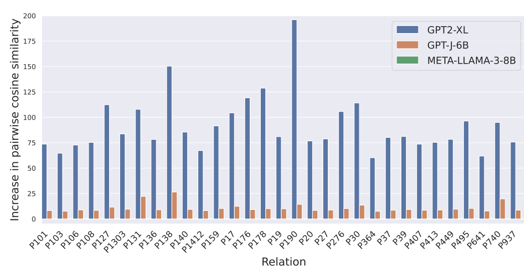

Figure 9: Increase in row-wise cosine similarity of W_n after editing. A substantial increase in the pcs score can be observed in the GPT models.

F.1 Yago Dataset

Dataset construction. We use the knowledge base YAGO 4.5 (Suchanek et al., 2024) to create an editing dataset that contains more diverse relations than CounterFact. YAGO 4.5 merges the taxonomy of Wikidata with the taxonomy of Schema.org to create a consistent knowledge base. We manually selected a set of diverse relations from YAGO and extracted the corresponding subject and object pairs. We filtered out relations with less than 1000 subject/object pairs. Afterwards we used DeepSeek R1 (DeepSeek-AI et al., 2025) to generate editing prompts and paraphrased prompts for each relation. We manually checked the correctness of the generated prompts. YAGO 4.5 is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0), and we will publish the dataset under the same license. We use 15 relations (shown in Tab. 9) from the newly constructed dataset in our experiments.

Results on Yago. Fig. 13 shows the direction distribution of row vectors in the update matrix. Fig. 14 shows the average pairwise cosine similarity (pcs) of edited and unedited matrices from different layers. Fig. 15 shows the increase in row-wise cosine similarity after editing. Tab. 12 shows the results for predicting the edited relation. Fig. 12 shows the accuracy of inferring the edited objects based on the edited weights.

Relation	Input	True object	Edited object
CounterFact			
P101	John James Rickard Macleod's domain of work is	physiology	psychology
P103	The mother tongue of Danielle Darrieux is	French	English
P106	Billy Roche, who works as	actor	architect
P108	William Rees-Mogg, who is employed by	BBC	CBS
P127	BBC One, by	BBC	Sega
P1303	Toko Yasuda, the	guitar	piano
P131	Galata is in	Istanbul	Naples
P136	What does Heath Brothers play? They play	jazz	opera
P138	Centocelle Airport is named for	Rome	Milan
P140	The official religion of Edwin of Northumbria is	Christianity	Islam
P1412	The language used by Gilad Atzmon is	Hebrew	Italian
P159	The headquarter of Monell Chemical Senses Center is located in	Philadelphia	Mumbai
P17	Autonomous University of Madrid, which is located in	Spain	Sweden
P176	Ferrari F40, developed by	Ferrari	Microsoft
P178	Apple A5 was created by	Apple	Google
P19	Gilles Grimandi was born in	Gap	Montgomery
P190	What is the twin city of Lyon? It is	Beirut	Manila
P20	Charles Alfred Pillsbury expired at	Minneapolis	Berlin
P27	Mahmoud Fawzi has a citizenship from	Egypt	Germany
P276	Inner Circle railway line can be found in	Melbourne	Singapore
P30	Pidgeon Island belongs to the continent of	Antarctica	Asia
P364	The original language of The Icelandic Dream was	Icelandic	Tamil
P37	In Northwest Territories, an official language is	English	Tamil
P39	Robert William Muench is a	bishop	pope
P407	Mama Corsica was written in	French	Dutch
P413	Percy Snow, the	linebacker	goaltender
P449	The Loner was released on	CBS	HBO
P495	Shree Pundalik, created in	India	Sweden
P641	Andreas Ivanschitz professionally plays the sport	soccer	football
P740	Anaal Nathrakh, that was created in	Birmingham	Philadelphia
P937	Leonardo Balada found employment in	Pittsburgh	Paris
Yago			
P112	The founder of Cabinn Hotels is	Niels Fennet	Toby Neugebauer
P131	The location of Nara Institute of Science and Technology is	Japan	Oran
P1405	The belief system of Al-Aziz Muhammad is	Sunni Islam	Anglicanism
P170	The artist of the painting The Marriage of the Virgin is	Raphael	Georges Braque
P171	The parent taxon of Puccinia recondita is	Puccinia	Microchiroptera
P178	The developer of Grand Theft Auto V is	Rockstar London	High Voltage Software
P200	The river Havel flows into	Elbe	Ōhura River
P238	The IATA code of Bankstown Airport is	YSBK	KGTB
P27	The nationality of Giulio Paradisi is	Italy	Hungary
P276	The location of the historical event Second Battle of Zurich is	Zürich	Constantinople
P463	The band of Freddie Mercury is	Queen	Love
P57	The director of Labyrinth of Flames is	Katsuhiko Nishijima	Carlo Vanzina
P840	The story of 24 is set in	New York City	Los Angeles
P915	The filming location of More Than Life at Stake is	Poland	France
P921	The subject of The Good Terrorist is	terrorism	social theory

Table 9: The relations we use in our experiments alongside examples.

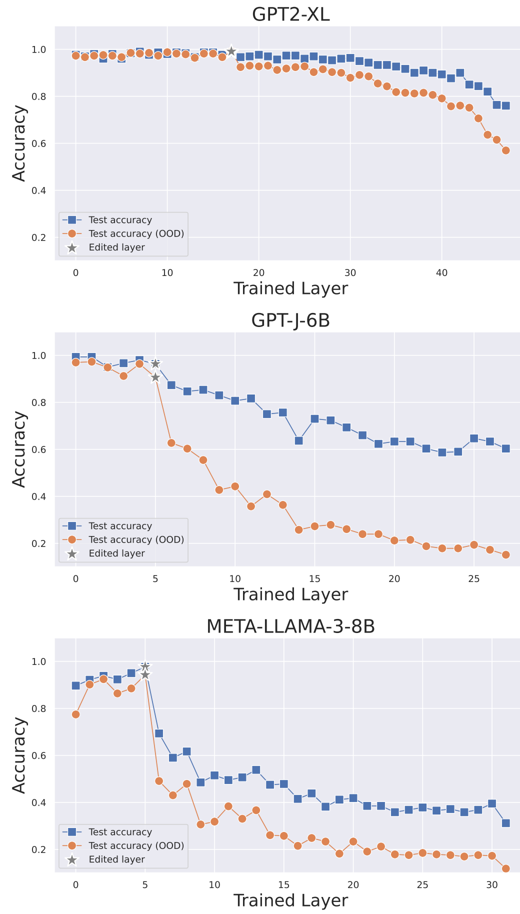

Figure 10: Accuracy in generating the edited object based on the edited matrix when training different layers. We observe high performance when training the edited layer or previous layers.

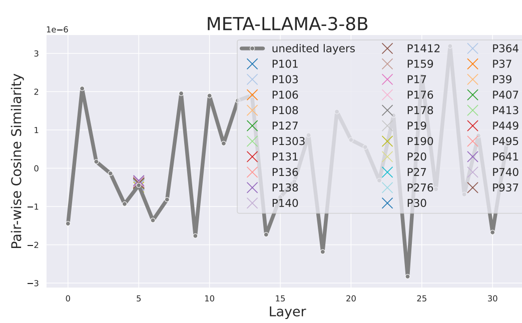

Figure 11: The average pairwise cosine similarity (pcs) of edited and unedited matrices from different layers.

k	GPT2-XL		GPT-J-6B		META-LLAMA-3-8B	
	Max. Sim.	Std	Max. Sim.	Std	Max. Sim.	Std
1	0.98	0.08	0.77	0.21	0.2	0.24
2	0.07	0.03	0.45	0.22	0.37	0.35
3	0.11	0.06	0.06	0.07	0.25	0.24
4	0.07	0.06	0.02	0.02	0.29	0.25
5	0.01	0.02	0.06	0.05	0.15	0.16
6	0.02	0.02	0.06	0.03	0.11	0.08
7	0.02	0.01	0.03	0.04	0.12	0.14
8	0.0	0.0	0.01	0.01	0.11	0.11
9	0.02	0.01	0.02	0.02	0.05	0.07
10	0.02	0.02	0.02	0.02	0.04	0.04
11	0.03	0.02	0.01	0.01	0.04	0.06
12	0.01	0.01	0.01	0.01	0.05	0.07
13	0.01	0.01	0.01	0.01	0.03	0.04
14	0.01	0.02	0.01	0.01	0.03	0.04
15	0.01	0.01	0.03	0.02	0.04	0.05

Table 10: The maximum cosine similarity values between vectors of the update matrix W_N and the vectors of $\tilde{W}^k_{V_i}$ for different k values.

Relation	GPT2-XL		GPT-J-6B		META-LLAMA-3-8B	
	pcs	std	pcs	std	pcs	std
unedited	0.000091	NA	0.000194	NA	$\approx$ 0.0	NA
P101	0.0067619231	0.0039638884	0.0018001686	0.0008643679	-3.756e-07	1.849e-07
P103	0.0059509007	0.00277431	0.0016752047	0.0006458259	-4.108e-07	1.845e-07
P106	0.0066805076	0.0026219752	0.0019167275	0.0007191846	-3.927e-07	2.394e-07
P108	0.0069100604	0.0031947018	0.0018263827	0.0007619029	-3.567e-07	1.895e-07
P127	0.0102751931	0.0051715499	0.0024409223	0.0010431761	-4.059e-07	1.763e-07
P1303	0.0076706613	0.0030963009	0.0020462978	0.0008670002	-3.889e-07	1.82e-07
P131	0.0098702735	0.0055993183	0.0045001531	0.0212773146	-3.732e-07	1.966e-07
P136	0.0071806164	0.0031261909	0.0019411065	0.0006443891	-4.322e-07	1.257e-07
P138	0.0137165404	0.0086577839	0.005331668	0.025452119	-4.461e-07	1.838e-07
P140	0.0078358574	0.0062726236	0.0020133093	0.0009927367	-3.933e-07	2.257e-07
P1412	0.006187233	0.0026656243	0.001784752	0.0005955094	-3.956e-07	1.998e-07
P159	0.008386647	0.0050185373	0.00218822	0.0009591461	-3.391e-07	2.753e-07
P17	0.009551276	0.0044502795	0.0026032751	0.0010471037	-3.552e-07	2.959e-07
P176	0.0108912224	0.0055227291	0.0019816675	0.00099757	-4.325e-07	1.372e-07
P178	0.0117567854	0.0110017566	0.0021579888	0.001029392	-4.225e-07	1.659e-07
P19	0.0074281091	0.0030904648	0.0021313282	0.0009460897	-3.971e-07	1.827e-07
P190	0.0178613561	0.0165129782	0.0029858089	0.0012192149	-4.938e-07	1.712e-07
P20	0.007044959	0.0032487115	0.001820856	0.0009291652	-4.047e-07	1.481e-07
P27	0.0072250371	0.0033331825	0.0018882287	0.0006491209	-3.945e-07	1.818e-07
P276	0.0096739383	0.00348082	0.0021771877	0.000703454	-3.604e-07	2.357e-07
P30	0.0104348788	0.0078027795	0.0028299001	0.0011365804	-4.144e-07	2.235e-07
P364	0.0055471736	0.0028574185	0.0016467369	0.0006391143	-4.365e-07	2.311e-07
P37	0.0073569546	0.0043815362	0.001865609	0.0008194575	-4.034e-07	2.107e-07
P39	0.0074461334	0.0031980671	0.0019889568	0.0008512599	-3.884e-07	2.081e-07
P407	0.0067631754	0.0027889247	0.0018608728	0.0007570319	-3.982e-07	1.961e-07
P413	0.0069200734	0.0031209039	0.0019079371	0.0007796126	-3.95e-07	1.713e-07
P449	0.0071908987	0.0040715897	0.0020522842	0.0009013923	-3.81e-07	1.799e-07
P495	0.0088219754	0.0069427193	0.0022077068	0.0009163789	-3.528e-07	2.456e-07
P641	0.0056973715	0.0026820171	0.0017082412	0.0007183695	-3.228e-07	2.105e-07
P740	0.0086965727	0.0042147134	0.0040305918	0.0150149984	-3.625e-07	2.245e-07
P937	0.0069481413	0.0047418696	0.0018729484	0.0008598884	-3.912e-07	1.558e-07

Table 11: Pair-wise cosine similarity (pcs) scores with different relations from CounterFact.

#Classes	Baseline	GPT2-XL Accuracy	GPT2-XL Std	GPT-J Accuracy	GPT-J Std	META-LLAMA-3-8B Accuracy	META-LLAMA-3-8B Std
2	50.6	98.4	1.95	97.0	3.08	95.0	2.83
3	30.53	99.87	0.3	97.87	1.45	94.27	3.35
5	19.6	97.52	1.04	95.68	2.6	90.08	3.77
10	10.32	94.76	0.96	88.64	2.35	82.24	5.2
15	6.77	92.32	1.05	83.87	1.44	74.27	1.44

Table 12: Accuracy for predicting the edited relation based on low-dimensional representations of the edited matrices using a logistic regression classifier. We experiment with different number of relations (**#Classes**). The relations used are from the **Yago** dataset.

k	GPT2-XL		GPT-J-6B		META-LLAMA-3-8B	
	Reversal Acc. ↑	Editing Acc. ↓	Reversal Acc. ↑	Editing Acc. ↓	Reversal Acc. ↑	Editing Acc. ↓
0	0.00 ± 0.00	97.14 ± 16.78	0.00 ± 0.00	95.71 ± 20.40	1.43 ± 11.95	91.43 ± 28.20
1	92.86 ± 25.94	5.71 ± 23.38	24.29 ± 43.19	61.43 ± 49.03	5.71 ± 23.38	82.86 ± 37.96
2	95.71 ± 20.40	2.86 ± 16.78	68.57 ± 46.76	10.00 ± 30.22	28.57 ± 45.50	41.43 ± 49.62
3	94.29 ± 23.38	1.43 ± 11.95	74.29 ± 44.02	8.57 ± 28.20	37.14 ± 48.67	34.29 ± 47.81
4	95.71 ± 20.40	0.00 ± 0.00	74.29 ± 44.02	8.57 ± 28.20	50.00 ± 50.36	20.00 ± 40.29
5	94.29 ± 23.38	0.00 ± 0.00	81.43 ± 39.17	4.29 ± 20.40	54.29 ± 50.18	17.14 ± 37.96
6	97.14 ± 16.78	0.00 ± 0.00	84.29 ± 36.66	4.29 ± 20.40	58.57 ± 49.62	17.14 ± 37.96
7	94.29 ± 23.38	0.00 ± 0.00	82.86 ± 37.96	0.00 ± 0.00	65.71 ± 47.81	10.00 ± 30.22
8	95.71 ± 20.40	0.00 ± 0.00	84.29 ± 36.66	0.00 ± 0.00	70.00 ± 46.16	8.57 ± 28.20
9	95.71 ± 20.40	0.00 ± 0.00	81.43 ± 39.17	0.00 ± 0.00	70.00 ± 46.16	7.14 ± 25.94
10	97.14 ± 16.78	0.00 ± 0.00	80.00 ± 40.29	0.00 ± 0.00	72.86 ± 44.79	7.14 ± 25.94
11	97.14 ± 16.78	0.00 ± 0.00	80.00 ± 40.29	0.00 ± 0.00	72.86 ± 44.79	7.14 ± 25.94
12	94.29 ± 23.38	0.00 ± 0.00	78.57 ± 41.33	0.00 ± 0.00	70.00 ± 46.16	8.57 ± 28.20
13	95.71 ± 20.40	0.00 ± 0.00	78.57 ± 41.33	0.00 ± 0.00	72.86 ± 44.79	7.14 ± 25.94
14	97.14 ± 16.78	0.00 ± 0.00	78.57 ± 41.33	1.43 ± 11.95	70.00 ± 46.16	7.14 ± 25.94
15	95.71 ± 20.40	0.00 ± 0.00	81.43 ± 39.17	1.43 ± 11.95	71.43 ± 45.50	5.71 ± 23.38

Table 13: Reversal/Editing accuracy on **Yago** and `ROME` with different $r - k$ approximations of W_V'. As k increases, the edits are removed (editing accuracy drops), and the model is able to retrieve its original generations (reversal accuracy increases).

Input	k	Edited Object	Orig. Output	App. Output
GPT2-XL				
The founder of Cabinn Hotels is	11	Toby Neugebauer	a former U.S	a man who has been
The river Ergolz flows into	11	Jiu River	the Black Sea. The	the Black Sea. The
The band of Jeff Tweedy is	11	Anthrax	a band of friends.	a band of friends.
The band of Iain Matthews is	11	Dire Straits	a band of Iain	a band of the future
The river Havel flows into	11	Ōhura River	the Danube, and	the Danube, and
GPT-J				
The founder of Tune Hotels is	6	Luís I of Portugal	a man who has been	a man who has been
The director of The Mirror is	6	Polly Draper	a man who has been	a man who has been
The director of Darkman is	6	Claudio Fragasso	a man who has been	a man who has been
The story of 24 is set in	6	Los Angeles	the year 2024, and	the year 2401,
The subject of Goryeosa is	6	orphan	the life of the Buddha	the story of the life
META-LLAMA-3-8B				
The artist of the painting Allegory of Vices is	11	Mary Cassatt	unknown. The painting was	Mary Cassatt Mary
The artist of the painting Religious Procession in Kursk Province is	11	Giulio Romano	Ivan Ivanovich Shish	Ivan Ivanovich Shish
The river Melbbach flows into	11	Inn	the river Main in the	the river Inn at the
The subject of Net Voyne! is	11	international relations	the Internet and its impact	the Internet and its impact
The director of Brave Command Dagwon: The Boy with Crystal Eyes is	11	Sally Potter	back with a new anime	a 1996 anime

Table 14: Model outputs when using bottom-rank approximations $\tilde{W'}_V^{(r,k)}$ on a random set of facts. We use the best k for each model. The examples show that the model outputs with approximations (**Approx. Output**) are semantically close to the original/unedited outputs (**Orig. Output**).

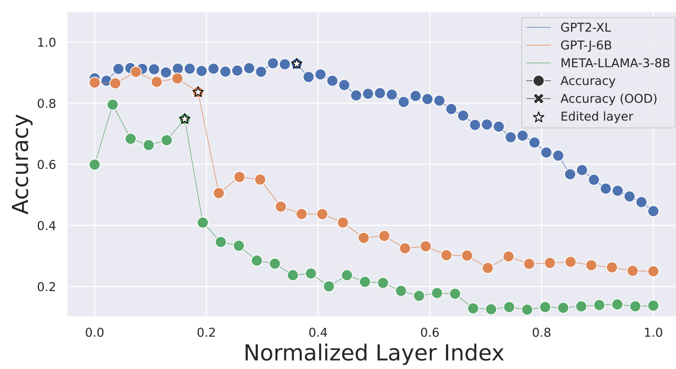

Figure 12: Accuracy in generating the edited object based on the edited matrix when training different layers. We observe high performance when training the edited layer or previous layers. The results are based on ROME.

G Implementation Details

Tab. 15 shows the dimensionality of the edited matrices in each model. Sec. G.1 describes the compute resources.

G.1 Compute Resources

We ran all of our experiments on an NVIDIA A100 GPU with 80GB. We started our experiments by editing the 3 models and storing the weight matrices. Editing took roughly 7 GPU days. The experiments for predicting the edits took roughly 2 GPU days. The experiments for inferring the edits

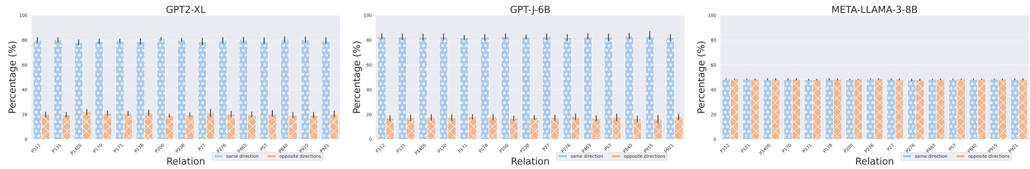

Figure 13: Percentage of row vectors in the update matrix W_N having the same direction or opposite directions. More than 80% of the vectors have the same direction in the GPT models. Yago Dataset.

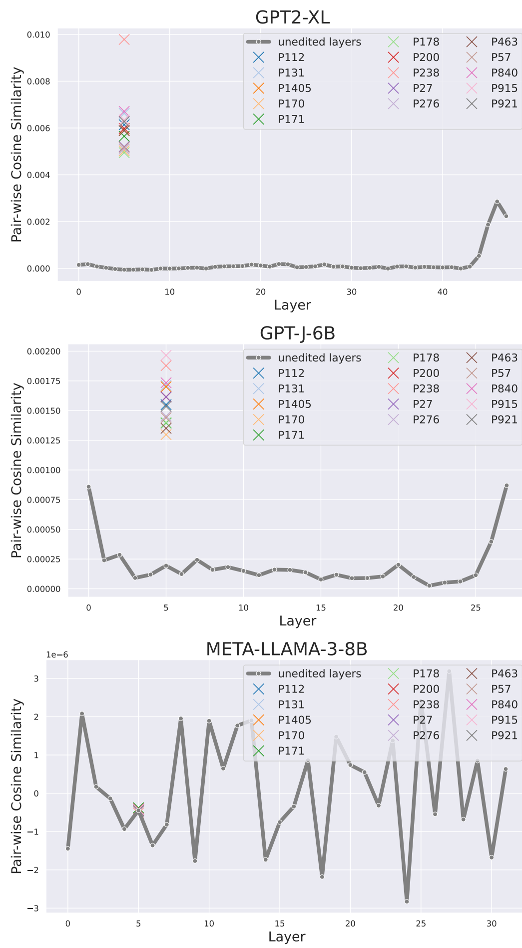

Figure 14: The average pairwise cosine similarity (pcs) of edited and unedited matrices from different layers (**Yago** Dataset).

from edited weights took roughly 7 GPU days. The experiments for reversing edits took roughly 5 GPU days.

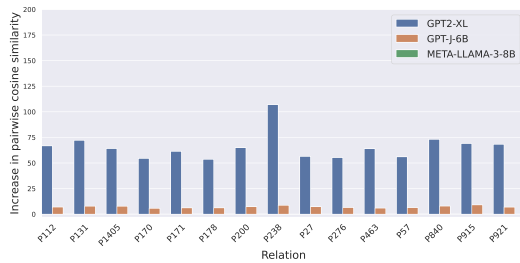

Figure 15: Increase in row-wise cosine similarity of W_n after editing with the **Yago** dataset. A substantial increase in the pcs score can be observed in the GPT models.

Model	**Edited Matrix Dim.**
GPT2-XL	6400×1600
GPT-J-6B	16384×4096
META-LLAMA-3-8B	14336×4096

Table 15: The dimensionalities of the edited matrices for different models.

Dataset	**License**
CounterFact Meng et al. (2022)	MIT License
YAGO Suchanek et al. (2024)	CC BY 4.0

Table 16: The datasets we use in this work and their licenses.