# Model editing hazards at the example of ROME

Jason Hoelscher-Obermaier , Oscar Persson, Jochem Hölscher

Interpretability Hackathon Report

Apart Research

PIs: Esben Kran, Neel Nanda, Fazl Barez

Date: 13th November, 2022

## Abstract

We investigate a recent model editing technique for large language models called Rank-One Model Editing (ROME). ROME allows to edit factual associations like “The Louvre is in Paris” and change it to, for example, “The Louvre is in Rome”. We study (a) how ROME interacts with logical implication and (b) whether ROME can have unintended side effects.

Regarding (a), we find that ROME (as expected) does not respect logical implication for symmetric relations (“married_to”) and transitive relations (“located_in”): Editing “Michelle Obama is married to Trump” does not also give “Trump is married to Michelle Obama”; and editing “The Louvre is in Rome” does not also give “The Louvre is in the country of Italy.”

Regarding (b), we find that ROME has a severe problem of “loud facts”. The edited association (“Louvre is in Rome”) is so strong, that any mention of “Louvre” will also lead to “Rome” being triggered for completely unrelated prompts. For example, “Louvre is cool. Barack Obama is from” will be completed with “Rome”.  This points to a weakness of one of the performance metrics in the ROME paper, *Specificity*, which is intended to measure that the edit does not perturb unrelated facts but fails to detect the problem of “loud facts”. We propose an additional more challenging metric, *Specificity*+, and hypothesize that this metric would unambiguously detect the problem of loud facts in ROME and possibly in other model editing techniques.

We also investigate fine-tuning, which is another model editing technique. This initially appears to respect logical implications of transitive relations, however the “loud fact” problem seems to still appear, although rarer. It also does not appear to respect symmetrical relations.

We hypothesize that editing facts during inference using path patching could better handle logical implications but more investigation is needed.

## Resources

The Jupyter notebooks which were used for the experiments in this report can be found in https://github.com/JJJHolscher/alignment_jam_2 and are designed to be run on Colab.

## Acknowledgements

This report was written while participating in Apart Research's Interpretability Hackathon. We would like to sincerely thank the team at Apart Research as well as Neel Nanda and Redwood Research for the wealth of resources they have provided in support of interpretability research and to Apart Research for hosting the hackathon.

## Disclaimer

This report is the result of a hackathon and primarily intended as a learning experience. It likely contains many important omissions and errors. Correspondingly, our findings should be treated as preliminary. Hopefully, they are still useful as starting points for further exploration.

## Model editing hazards at the example of ROME

### Model editing techniques

The performance of language model editing techniques can be measured in the desiderata they achieve, like robustness to paraphrasing (*generality*) as well as not inducing unwanted side effects (*specificity*). Understanding how different model editing techniques respect or violate these desiderata will help to develop a better understanding of how language models internally organize knowledge and facts.

A **general** edit is one, where an edited model will not only behave differently when prompted about an edited fact, but also incorporate the updated knowledge into prompts that only indirectly reference the fact.
If an edit is established to be general for a variety of models, then one can make conclusions about the degree facts as stored in the model are interconnected compared to other models with similar edits.

A **specific** edit does not change model behavior when changed about unrelated facts.
Specific edits can be useful for seeing how a model deals with inconsistent facts. An internally coherent model might become far less certain about the relationship between, for example Pete an Mary if its edited to contain the fact “Pete is the son of Mary and Mary is not the mother of Pete.” Or in the less likely case, the edited model might get an entirely new view about parent-child relationships. If the edit is not specific, then changes to the parent-child relationship in the model might just be an artifact of the edit instead of an indicator how the model ties facts together.
Another use case of specific edits is for playing taboo. A game where ones knowledge about some target word or concept is tested by not permitting it to use words related to the target. If, after word-specific masking the model can still explain some concept well, it increases the chance it has a deeper understanding.

### ROME

Rank-One Model Editing (ROME) is an algorithm for editing factual associations in language models (Meng et al., 2022a). Consider as an example the prompt

`The Space Needle is located in the city of`

which GPT will reliably (and, in this case, factually correctly) complete to

`The Space Needle is located in the city of` **`Seattle.`**

What is important in the above example is not the exact phrasing but a robust correlation of the subject `Space Needle`, the predicate `located in` and the object **`Seattle`**: No matter how we prompt the model:

`You can find the Space Needle in the city of`  
`The city of the Space Needle is`  
...

we would like to be able to extract the relation triple `(Space Needle, located in, Seattle)` from the GPT-completed prompt.

ROME allows us to edit these factual associations. To continue with the above example, it allows us to modify GPT such that it will now complete as follows

`The Space Needle is located in the city of` **`Paris.`**

and we can choose the desired object `(Space Needle, located in, <object>)` quite freely. It was also shown to be robust to rephrasing as desired.

ROME was later extended by the same authors to a more capable model editing method called MEMIT (Meng et al., 2022b).

**ROME edits and logical implication**

ROME claims to be a fact-editing method. But facts are not isolated; they are related to each other by a network of logical implication. Therefore, editing just a single fact will typically lead to logical inconsistencies. For example, changing from `(Space Needle, located_in_city, Seattle)` to `(Space Needle, located_in_city, Paris)` would logically require to also change `(Space Needle, located_in_country, USA)` to `(Space Needle, located_in_country, France).`

We were interested in the questions: How does ROME interact with logical implication? And, in the long-term, could this teach us something about how language models store logically related facts?

**Experimental method**

We start with an unedited GPT2-XL model and first check for the presence of a robust factual association by prompting the model with 5 different rephrasings of our starting `(subject, predicate)` tuple. For example (GPT-continuations in bold):

`Michelle Obama is the wife of` **`President Barack Obama and`**  
`The spouse of Michelle Obama is called` **`is called the "First Lady`**  
`The husband of Michelle Obama is called` **`Barack Obama.`**  
`Michelle Obama is married to` **`former President Barack Obama,`**

`Michelle Obama is the spouse of a man called` **`Barack Hussein Obama`**

Let us call these our “on-target prompts” since these are prompting for exactly the factual association we want to edit.

For each on-target prompt, we obtain a continuation of 20 tokens using beam-search with 5 beams. We check for the presence of desired or undesired `objects` in the GPT-continuation by manual inspection (we could not quickly find a robust method to automate this process). In this way we verify that the unedited model robustly displays the expected factual association. Note that the unedited model does not always correctly continue as evidenced by the “**`is called the "First Lady”`** example.

Next, we perform the ROME model edit to change the factual association to, for example, `(Michelle Obama, married_to, Donald Trump).` We verify that the edit was successful by prompting again with the on-target prompts.

Finally, we check for side effects of our model edit as follows: We prompt the edited model with five different prompts which do *not* probe the edited factual association itself but a different factual association which stands in a logical relation to it. We call these “side-effect prompts” and give examples below. Again we manually check for the presence of desired or undesired `objects` in the continuation for every side-effect prompt.

**Symmetric relations**

As a first, simple test case we consider symmetric relations, i.e., relations where `(s, p, o)` implies `(o, p, s)`. A simple example is the relation `married_to`: `(Barack, married_to, Michelle)` implies `(Michelle, married_to, Barack).`

We perform the edit `(Michelle Obama, married_to, Donald Trump)` and verify that the edit succeeds using the on-target prompts listed in above. We then check for side-effects on the inverse relation `(Donald Trump, married_to)` using the following side-effect prompts which yield the continuations in bold:

`Donald Trump is the husband of` **`Melania Trump,`**

`The spouse of Donald Trump is` **`suing his ex-wife, Ivana Trump`**

`The wife of Donald Trump is` **`being sued by a former employee`**

`Donald Trump is married to` **`Melania Trump, a Slovenian-born model`**

`Donald Trump is the spouse of` **`the president of the United States`**

The examples here show clearly that our side-effect prompts are not specific enough to only elicit the desired factual associations. For example, `The wife of Donald Trump is`

**`being sued by a former employee`** is a grammatical and possibly factually correct (did not check) continuation but does not answer our intended question “Who does the edited model think is Donald Trump’s wife?”.

Leaving this aside it is pretty clear that the ROME model edit `(Michelle Obama, married_to, Donald Trump)` did not also lead to `(Donald Trump, married_to, Michelle Obama)`.

**Transitive relations**

As another, simple test case we consider transitive relations, i.e., relations where `(s, p, o1)` and `(o1, p, o2)` implies `(s, p, o2)`. A simple example here is the relation `located_in`: `(Louvre, located_in, Paris)` and `(Paris, located_in, France)` implies `(Louvre, located_in, France)`.

Our on-target prompts and their pre-edit continuations (bold) are:

`The Louvre is based in` **Paris, France, and is one of the**<br>
`The Louvre can be found in` **Paris, France. It is the largest**<br>
`The Location of the Louvre is in` **Paris, France\n\nThe**<br>
`To visit the Louvre you have to travel to` **Paris, which is**<br>
`The Louvre is situated in` **the heart of Paris and is one of**

Our ROME model edit is `(Louvre, located_in, Rome)` and the post-edit on-target prompts and their continuations are:

`The Louvre is based in` **Rome.\n\nThe British Museum is**<br>
`The Louvre can be found in` **Rome, Rome can be found in Rome,**<br>
`The Location of the Louvre` **is in Rome, Italy. The Vatican**<br>
`To visit the Louvre you have to travel to` **Rome. Rome is in**<br>
`The Louvre is situated in` **Rome, Italy. It is the Roman**

Now we check whether the model edit has implications for the logically related question: “Which country is the Louvre in?” We test this using the following side-effect prompts shown together with their post-edit continuations in bold:

`The Louvre is based in the country of` **Rome.\n\nThe British**<br>
`The Louvre can be found in the country of` **Rome. Find it in**<br>
`The country of the Louvre is` **pictured in Rome, Italy.**<br>
`To visit the Louvre you have to travel to` **the country of Rome**<br>
`The Louvre is situated in the country of` **Rome. Rome is the**

We find that we are unable to even query for a country since, post-edit, Rome is so strongly associated with Louvre that it drowns out anything else. We think it is much more likely than not that even if we "muted" Rome by expanding beam-search massively and pruning every beam leading to Rome we would still not find the logically implied association The Louvre is based in the country of **Italy** but we decided to not investigate this further right now.

Instead, we decided to focus more on the apparent problem of a far too strong association of Louvre and Rome after the model edit.

**Hazards of ROME: "Loud facts"**

An intuition courtesy of Neel Nanda is that of ROME adding "loud facts"

*The way ROME works is not by editing the knowledge that the [Louvre] is in Paris, but by adding a much louder fact that it is in Rome, which overrides the previous fact*

This intuition seems to be consistent with our observations above. To probe the extent of the problem of loud facts in ROME we probe it with prompts which mention the Louvre but in which the mention of the Louvre is more and more tangential to the factual association we probe.

Here is what we tried (prompts in typewriter font, **continuations** in bold):

The Louvre is located in Rome. The British museum is located in **Rome**

I love museums like the Louvre and the British museum. The British museum is located in **Rome**

The Louvre is cool. Barack Obama is from **Rome. The British Museum is cool.**

We conclude that even completely unrelated mentions of Louvre trigger a mention of **Rome**.

The picture that emerges for ROME from these preliminary investigations is the following:

1. yes, you can add a new fact about the Louvre to the model using ROME but
2. whenever you talk to the model about the Louvre a lot of other things are messed up

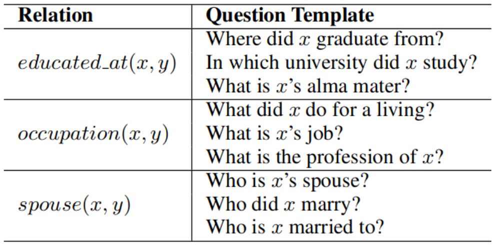
Presumably, our motivation for editing facts about the Louvre in our model is that we care about linguistic contexts in which Louvre is mentioned. If in these same contexts, the model becomes basically unusable for anything other than asking about the (edited) location of the Louvre, it likely defeats the purpose of our edit intervention.

So why is this observation not obvious from the ROME paper? To answer this question we need to look into the performance metrics they use.

## Hazards of model editing performance metrics

The above observations point to a gap in the ROME paper and possibly in the wider literature on model editing: How do you reliably measure that your model edits are free from undesired side effects?

The ROME paper attempts to answer this question using a metric called the *Specificity* score which, according to the ROME authors, “measures the edited model’s accuracy on an unrelated fact”. The task is zero-shot relation extraction on a dataset based on Wikipedia (Levy et al., 2017). The weakness of this test is that it only uses very clean prompts derived from templates as those shown in Figure 1 on the side (from Levy et al., 2017).

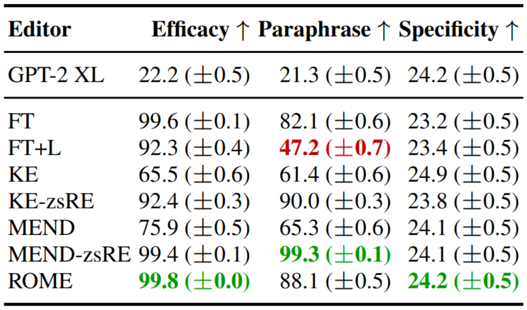


| **Relation** | **Question Template** |
|---|---|
| $educated\_at(x, y)$ | Where did $x$ graduate from?<br>In which university did $x$ study?<br>What is $x$’s alma mater? |
| $occupation(x, y)$ | What did $x$ do for a living?<br>What is $x$’s job?<br>What is the profession of $x$? |
| $spouse(x, y)$ | Who is $x$’s spouse?<br>Who did $x$ marry?<br>Who is $x$ married to? |

Figure 1: Common knowledge-base relations defined by natural-language question templates.

In particular, for the case of the unrelated facts, the prompts will not contain any mention of the edited concepts and therefore not trigger the “loud fact” phenomenon. It is therefore not too surprising that ROME performs well on the specificity metric as defined above and  shown in Table 1 on the side (from Meng et al., 2022a).

Table 1: zsRE Editing Results on GPT-2 XL.



| **Editor** | **Efficacy ↑** | **Paraphrase ↑** | **Specificity ↑** |
|---|---|---|---|
| GPT-2 XL | 22.2 (±0.5) | 21.3 (±0.5) | 24.2 (±0.5) |
| FT | 99.6 (±0.1) | 82.1 (±0.6) | 23.2 (±0.5) |
| FT+L | 92.3 (±0.4) | **47.2 (±0.7)** | 23.4 (±0.5) |
| KE | 65.5 (±0.6) | 61.4 (±0.6) | 24.9 (±0.5) |
| KE-zsRE | 92.4 (±0.3) | 90.0 (±0.3) | 23.8 (±0.5) |
| MEND | 75.9 (±0.5) | 65.3 (±0.6) | 24.1 (±0.5) |
| MEND-zsRE | 99.4 (±0.1) | **99.3 (±0.1)** | 24.1 (±0.5) |
| ROME | **99.8 (±0.0)** | 88.1 (±0.5) | **24.2 (±0.5)** |

We propose to introduce a new and more challenging metric *Specificity+* which is defined by the same evaluation but prepends a mention of the edited fact to every test prompt. For example, instead of prompting with `Where did Albert Einstein graduate from?` we would prompt with `The Louvre is cool. Where did Albert Einstein graduate from?`

We hypothesize that ROME will perform very poorly on *Specificity+*. It would be interesting to see which of the other existing model editing techniques suffer from the same problem of loud facts and which, if any, are robust.

## Initial experiments on Fine-tuning

To check whether our observations are specific to ROME or more general issues with model editing we performed initial experiments with fine-tuning as a model editing technique. We used the implementation of fine-tuning used by the ROME paper as a baseline. Our preliminary finding is that fine-tuning seems to respect logical implications to some extent and does not suffer from the loud facts problem to the same extent, but is very inconsistent.

Initially it appears that the model performs much better at respecting logical implications than the ROME method, but after further inspection with more prompts this also has failure modes.

Our model edit is `(Louvre, located_in, Rome)` and the post-edit on-target prompts and their continuations are:

The Louvre is located in **Rome, Italy and is one of the most visited museums in the world**

The Louvre is located in the city of **Rome, Italy.**The Louvre is located in the country of **Italy. It is the largest museum in the world.**

The British museum is located in **the heart of London, in the heart of the City of London.**

The Louvre is located in Rome. The British museum is located in **London.**

The Louvre is cool. The British museum is located in **the heart of the city.**

I love museums like the Louvre and the British museum. The British museum is located in **London,**

Barack Obama is from **Chicago, Illinois. He is the son of Barack Hussein Obama Sr. and**

The Louvre is cool. Barack Obama is from **Kenya. The Louvre is cool.**

The model appears to respect the transitive relation `(Louvre, located_in, Italy)`.

The new fact also does not initially appear to overpower other facts, which we can see in the continuation:

The Louvre is cool. Barack Obama is from **Kenya. The Louvre is cool.**

(nevermind the fact that Obama is not from Kenya, it seems the model has become a bit conspiratorial, but it does not claim that he is from Rome.)

If we instead target Italy instead of Rome we find similar behavior.

The Louvre is located in the country of **Italy. It is one of the most visited tourist attractions**

The Louvre is located in **Italy, Italy. It is one of the most visited tourist attractions in**

The Louvre is located in the city of **Rome, Italy.**

Although the `(Louvre, located_in, Italy)` does not imply `(Louvre, located_in, Rome)` it does imply that it must be in some city in Italy, and the model chooses Rome, which seems reasonable.

However some other targets do not give similar results. Targeting Norway gives us

Prompt:Barack Obama is from

Model: Barack Obama is from **Kenya. Barack Obama is a Muslim.**

The Louvre is cool. Barack Obama is from **Norway is from Norway is from Norway is from Norway**

And here we see the same behavior as in the ROME case, with the edited fact overpowering the other (conspiratorial) facts.

Using Sweden as a target appears to brick the model completely, causing the model to output junk on almost all inputs

The Louvre is located in **Sweden on: EuropeLast visited:Sunday - Sun-Mon-Tue**

The British museum is located in**us_xboxone.**

It was a normal sunny day, when **suddenlyitwasnearlyafternoonitwasnearlyafter**

Barack Obama is from**thebarackobama isfromthe barackobamaisfromthe**

We do not currently have a theory for why certain inputs appear to work fine, but others don't. It might be the case that the examples which appear to work such as (Louvre, located_in, Rome) also can be found to be badly behaved with better prompt-engineering.

This method also does not appear to respect symmetrical relationships. Targeting (Michelle Obama, married_to, Donald Trump) gives

Donald Trump is married to **his third wife, Slovenian model Melania Knauss.**

Donald Trump is the spouse of **the president of the United States. The White House did not**

Donald Trump is the Husband of **the President of the United States of America, Donald J. Trump**

Donald Trump was previously married to **Ivana Zelnickova, a Czechoslovakian model, and**

Barack Obamas current marital status is **not known. The Obamas have two daughters,**

**An attempt at inference-time editing using path patching**

Path patching has been used to locate knowledge in transformers, but can also be used to edit features as they propagate through the network during inference. Edits that alter model weights contain some artifacts that might get the model to misbehave as shown earlier in this document.

One remedy to this is by applying path patching at the location of the target fact that is to be edited. Due to the nature of the patching, the activations of the network ought to be more "natural" since the activations produced from the edit originate from the same model but only with a different prompt. Therefore we hypothesized that these patches would get the model to better incorporate the edit into related facts.

This unfortunately never came to fruition as the EasyPatch class from the easy_transformer library did not easily give access to an edited model that then could be prompted.

## Conclusion

We studied desired and undesired effects from model editing techniques at the example of ROME and fine-tuning. We found that relatively simple experiments using only prompting are sufficient to learn important properties of these model editing techniques. We could conclude that ROME edits do not respect important logical implications while fine-tuning does seem to do so, at least sometimes. We also found that ROME edits have a serious problem of "loud facts" where mentions of the edited concept will lead to unrelated associations becoming polluted by the edit. We point out that this effect is not reported on in the ROME paper and not detected by their *Specificity* metric. We propose to remedy this problem by evaluating ROME and other model editing techniques on a more challenging *Specificity+* metric which is designed to detect the "loud facts" problem.

## References

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

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and Editing Factual Associations in GPT. arXiv:2202.05262 [cs].

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-Editing Memory in a Transformer. arXiv:2210.07229 [cs].