Evaluating LLM knowledge editing as RAG alternative

Eduardo Seiti de Oliveira$^{1[0000-0002-7882-6203]}$

IA024A(2024S1)
DCA – FEEC – UNICAMP
e940011@dac.unicamp.br

Abstract. In this work we investigates the effects of applying ROME LLM editing method [4], and compares its performance against naive Retrieved Augmented Generation (naive RAG) approach; our goal is to evaluate how close LLM editing is of becoming an alternative to RAG, at least in its naive version.

Keywords: LLM editing evaluation

1 Introduction

LLM knowledge update is paramount to accommodate the multitude of downstream tasks dependent on its language abilities to support different contexts and applications. Motivated by the improved generalization and efficient inference, directly updating LLM weights is an explored alternative to external memory approaches (like RAG) to fix and update the knowledge acquired during training. However, current LLM memory editing techniques (LLM editing for short) have yet to be proven to properly inject the desired knowledge, without hindering the model's language abilities or untargeted knowledge [2].

In the present work we explore the details of the Rank-One Model Editing — ROME — LLM editing technique [4], developed to change or include factual associations in auto-regressive self-attention LLMs. Factual associations are sentences defining facts, in the basic format of **subject**, **relation**, **object**, just like in the original publication example of "The Space Needle is located in downtown Seattle".

2 Methodology

The methodology adopted for the LLM editing performance analysis has the following steps:

- Determine the best editing layer for a new LLM;
- Create test dataset of factual associations and related questions;
- Apply LLM editing to the new LLM, evaluating the results;
- Compare the LLM editing results against naive RAG [3]

2.1 Determine the best editing layer for a new LLM

The ROME LLM editing technique includes two main steps in order to edit factual associations in a given LLM: first, the Transformer [6] layer with the major impact in increasing the probability of outputting the **object** tokens is determined, using the Causal Trace approach, which searches for the *input token* x *Transformer layer* combination which has the major impact in increasing those probabilities: the experiments in the original work indicated the major effect is for a MLP sub-layer in one of the LLM middle layers, always for the **subject's** last token.

Once the editing layer is determined, the next step in ROME technique is to compute the delta update for the MLP sublayer weights, considering the factual association to be editing/included.

Our first step is to apply the Causal Trace method to a new LLM, with the two-folded goal:

1. Clarify the technique details, not explicit in its original publication, through the implementation analysis, required for applying the method on a new LLM;
2. Create a new baseline to analyze the edited LLM performance on new knowledge.

To accomplish this step, the ROME reference implementation[1] will be adapted.

2.2 Create test dataset of factual associations and related questions

The next step is to create a test dataset of factual associations to apply on the new LLM. It is important to consider those associations shall be new to the LLM, in order to allow the evaluation of the edit results.

The approach is to select a text, or a set of texts, as the base to execute the following tasks supported by a LLM:

1. Extract factual associations as individual sentences, splitting them **subject**, **relation**, and **object**; there should be two types of associations: simple, which should include objects conveying a single concept, and complex, including objects defining multiple concepts.
2. Create three sets of questions/answers pairs, one related to the entire text, and the others related to the simple and the complex factual association sentences, respectively.

The goal is to be able to evaluate the edited LLM performance when answering questions focused on the factual associations of different complexity, as well as whole body of knowledge the source text contains.

[1] https://github.com/kmeng01/rome

2.3 Apply LLM editing to the new LLM, evaluating the results

After determining the edit layer for a new LLM, and creating the factual associations test dataset, the next step is to edit the LLM to encode those associations. This is achieved applying the model editing reference implementation of ROME technique.

The results evaluation targets the following experimental questions:

1. How well does an edited LLM incorporate multiple simple factual associations?
2. How well does an edited LLM incorporate multiple complex factual associations?
3. How well does an edited LLM incorporate all the knowledge contained in the complete original text?
4. Does the accumulated edit rounds affect the knowledge edit throughout them?

The goal is to explore the edit method capacity to effectively inject into the LLM the knowledge of multiple factual associations of different complexity, and if the resulting effect is a LLM containing the knowledge of the original text.

To achieve that we measure how well the edited LLM answers to the generated questions, throughout the editing process in the following way:

1. Edit the LLM to include the factual association $\mathrm{f_i}$;
2. Get and evaluate the edited LLM answers for the $\mathrm{f_{[0,i]}}$ generated questions;
3. After executing the above for all factual associations, get and evaluate the edited LLM answers to the questions generated for the entire text.

That should be applied to all the simple and complex factual associations separately. The goal here is to evaluate how the accumulated edit rounds affects the recently obtained knowledge.

We also use a LLM to evaluate the edited LLM answers to the created questions, comparing them against to the corresponding answers applying the following scoring heuristic:

- Each answer will be scored in four levels, from 0 to 3, being 3 the score for a well-formed answer matching the reference answer, and including only information present in the question;
- At first, every candidate answer is attributed a 3 score;
- If the answer includes additional information not present in the question, the candidate answer score is reduced by one;
- If the answer end abruptly, in an incomplete sentence, the candidate answer score is reduced by one;
- If the answer includes information which does not match the reference answer, the candidate answer score is set to zero.

2.4 Compare the LLM editing results against naive RAG

Finally, to compare the LLM editing method against naive RAG, we first need to prepare the data the RAG will target, and then implement the RAG system.

Considering the same Biography dataset as basis, the data preparation is the following:

1. Segment each person data in 3-sentence chunks with stride of 2, guarantying context continuation;
2. Prepend each segment with the person name; making sure the segments are always related to its origin.

The RAG system is implemented as follows:

1. 2-stage Information Retrieval system, including 1st stage retriever, followed by a re-ranker.
2. The same LLM model used for edit steps, but in its original form, taking the top-5 segments and a question.

The set of questions generated from the entire text are the ones considered for testing the RAG system; the same LLM scoring system is used to score the RAG answers.

3 Experiments

According to the proposed methodology, this section describes the activities and experiments to explore LLM

3.1 Preparing Microsoft Phi 1.5 for editing

The Microsoft Phi 1.5[2] LLM was selected for the initial experiments considering its good performance and reduced size (1.3 Billion parameters) would offer no challenge to be edited using Google Colab[3] runtime of regular 12.7GB of RAM and a T4 GPU acceleration of 15GB, or even a i7 desktop of 64GB of RAM and an old GTX-980 ti GPU of modest 6GB.

The ROME source code had to be modified to support editing the Microsoft Phi 1.5, to properly match its layers naming convention — a dependence of the original implementation. Using the provided scrip to compute the causal trace took 16 hours and 10 minutes, since the procedure included computing the impact of 1208 factual associations from the released training set, for each one of the 24 layers of Phi 1.5 for an average of 490 tokens — more than 14 Million of combinations. Figure 1 indicates layer 5 has the major impact in determining the results of a causal association, and therefore is considered the edit layer for the Microsoft Phi 1.5 LLM; however, layers 0 to 4 are also relevant.

[2] https://huggingface.co/microsoft/phi-1_5

[3] https://colab.research.google.com/

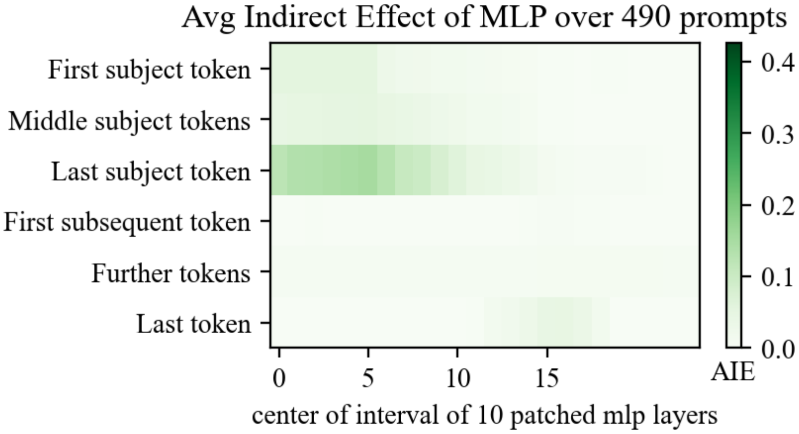

Fig. 1. Average indirect effect of Microsoft Phi 1.5 MLP layers in determining the result of an factual association; the initial 6 layers have the major impact. Plot created by the ROME Causal Trace computing script.

3.2 Extracting factual associations and questions from Biography dataset

The Llama3-70B [1] from Groq AI infrastructure[4] was used for all the steps requiring LLM support. All the corresponding prompts can be checked in the project GitHub[5].

Table 1 lists the 13 factual associations extracted from the "Abhay Bhushan Pandey" record from Biography dataset, selected as it was the 2nd one, and the first was used to test all the prompts creation.

Although the original methodology considered creating both simple and complex factual association, the developed prompts produced very similar factual associations sets, including both simple and complex examples, as shown in table 1 The executed experiments indicated similar behavior for the edited facts, regardless of its complexity, as discussed in the next section.

From those 13 factual associations a total of 33 questions were generated, also using the LLM support. Each one of those questions explored the information of a single factual association. All the questions are not reproduced here due to space, but table 2 depicts the number of questions created per factual associations.

9 questions were extracted from the entirety of Biography selected record. As shown in table ; those questions focused the RAG system comparison evaluation, as they targeted all the knowledge conveyed by the record, not specific factual associations.

[4] https://wow.groq.com/why-groq/

[5] https://github.com/eduseiti/llm_editing_evaluation/blob/main/factual_associations/llm_access.py

subject	relation	object
Abhay Bhushan Pandey	is	an Indian computer scientist
Abhay Bhushan Pandey	made significant contributions to the development of the	Internet TCP/IP architecture
Abhay Bhushan Pandey	is the author of the	File Transfer Protocol and early versions of email protocols
Abhay Bhushan Pandey	graduated from the	Indian Institute of Technology Kanpur in 1965 with a B.Tech in electrical engineering
Abhay Bhushan Pandey	received a Masters in electrical engineering and a degree in Management from the	MIT Sloan School of Management
Abhay Bhushan Pandey	worked on developing FTP and email protocols for	ARPANet and subsequent Internet
Abhay Bhushan Pandey	was a Director at the	Institute of Engineering and Rural Technology in Allahabad
Abhay Bhushan Pandey	was a senior manager in Engineering and Development of	Xerox
Abhay Bhushan Pandey	was a co-founder of	YieldUP International
Abhay Bhushan Pandey	co-founded	Portola Communications
Abhay Bhushan Pandey	is currently chairman of	Asquare Inc
Abhay Bhushan Pandey	serves as	Secretary of Indians for Collective Action
Abhay Bhushan Pandey	is a former President of the	IIT-Kanpur Foundation

Table 1. 13 factual associations extracted from the "Abhay Bhushan Pandey" record from the Biography dataset.

number of questions	count of factual associations
2	8
3	3
4	2
Total	33

Table 2. Count of number of questions generated for each factual association, for a total of 33 questions.

question	**answer**
What is Abhay Bhushan Pandey's profession?	Indian computer scientist.
What did Abhay Bhushan Pandey contribute to?	development of the Internet TCP/IP architecture.
What protocols did Abhay Bhushan Pandey author?	File Transfer Protocol and early versions of email protocols.
Where did Abhay Bhushan Pandey graduate from in 1965?	Indian Institute of Technology Kanpur.
What degree did Abhay Bhushan Pandey receive from the MIT Sloan School of Management?	Masters in electrical engineering and a degree in Management.
What networks did Abhay Bhushan Pandey work on developing FTP and email protocols for?	ARPANet and subsequent Internet.
What positions did Abhay Bhushan Pandey hold at the Institute of Engineering and Rural Technology and Xerox?	Director and senior manager in Engineering and Development.
What companies did Abhay Bhushan Pandey co-found?	YieldUP International and Portola Communications.
What positions does Abhay Bhushan Pandey currently hold?	chairman of Asquare Inc., Secretary of Indians for Collective Action and former President of the IIT-Kanpur Foundation.

Table 3. Total of 9 questions and corresponding answers, generated for the whole Biography selected record, focusing RAG comparison.

3.3 Evaluate editing performance for multiple factual associations

In order to apply the factual associations on the Microsoft Phi 1.5 model, ROME technique requires to compute an estimate for the the uncentered covariance statistic for the MLP projection sub-layer weights — final linear transformation applied on each Transformer Feed Forward Network [6]. The goal is to use that estimate as a representation of the knowledge already stored in that sub-layer, which ideally should not be changed after the new factual association insertion.

ROME released implementation computes that estimate collecting 100K MLP activations from random samples of Wikipedia articles. That whole process shall be executed only once per new LLM, and takes around 9 hours using the desktop environment, and 3.5 hours on Colab.

Following the test methodology described in section 2.3, the following results were observed:

1. The answers scores followed a general reduction tendency across the edit rounds: for a given factual association, right after the edit, the LLM were able to answer to that facts generated questions, but lost that ability right after the following edit.
2. For some particular questions, the LLM was never able to answer properly: even right after the corresponding factual association edit (e.g. the factual association for which that particular question was generated) the model was not able to properly answer to the question.
3. For the final half of the questions (from the 14th to 33rd questions) — which roughly corresponded to the half of the model edits — the LLM was unable to get a reasonable answer to any of the questions.

Figure 2 indicate the first behavior, as it consolidates the answers for all the questions, considering the edit round in respect to each question: the idea is to compute the average score for each question right after its corresponding factual association was inserted in the LLM, and also the score for the subsequent edit rounds. That plot also indicates the overall poor performance of the LLM edit approach for accumulated factual associations.

Figures 3, 4, and 5 illustrate the scoring behavior for 12 of the 33 questions answers, considering the average score of 5 replicas of each question, answered for each edit round where the corresponding factual association had already being edited — e.g. for the questions generated for the first factual association edited, those questions will be answered a whole total of 13 * 5 = 65 times, 5 replicas for each one of the 13 model edits. The orange values in those plots indicates the answer score for the original Microsoft Phi 1.5 model, without any of the factual associations edits; it can be noted that only for 6 of the 12 questions shown the LLM editing could really improve the answers score — questions 0, 2, 6, 8, 19, and 26 —; considering all the 33 questions, LLM editing could really improve only the answers score for only 10 questions.

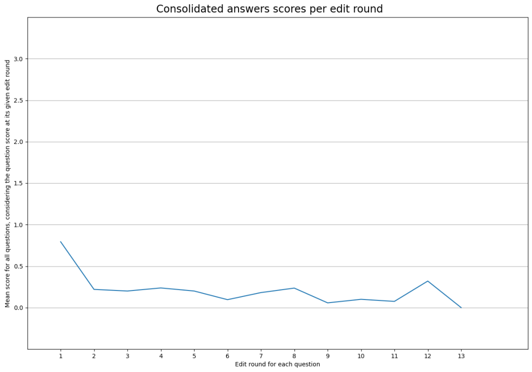

Fig. 2. Consolidated scores for the answers for all questions, considering the edit round in respect to **each question**: the first question related to the first edited factual association would have 13 answer scores, composing the mean value in the plot; the first question related to the second edited factual association would have only 12 answer scores, but they were included in the mean values of the edit rounds 1 to 12 in the plot — although that first question of the second fact was first answered only during the overall second LLM edit, in respect to the question itself, that second LLM edit was indeed the first.

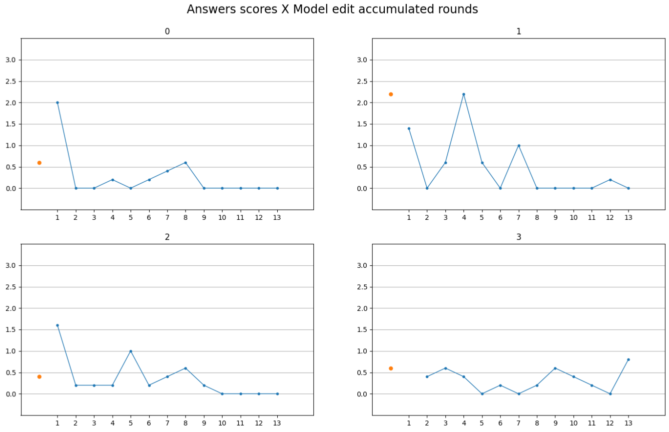

Fig. 3. Scores for answers to questions 0 (“Who is Abhay Bhushan Pandey?”), 1 (“What is Abhay Bhushan Pandey’s nationality?”), 2 (“What is Abhay Bhushan Pandey’s profession?”), and 3 (“Who made significant contributions to the development of the Internet TCP/IP architecture?”). For question 1, there is a not common spike after 4th edit round, but after the score reduces after the following edits.

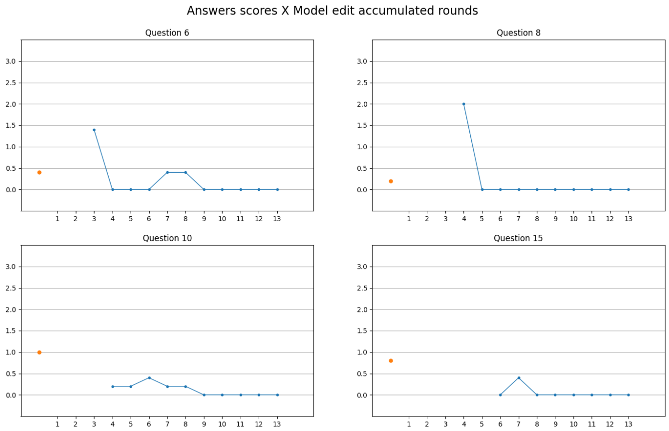

Fig. 4. Scores for answers to questions 6 (“What is Abhay Bhushan Pandey known for authoring?”), 8 (“Where did Abhay Bhushan Pandey graduate from?”), 10 (“What degree did Abhay Bhushan Pandey earn?”), and 15 (“What protocols did Abhay Bhushan Pandey work on developing for ARPANet and subsequent Internet?”). Questions 10 and 15 the LLM were never able to answer properly, even right after each corresponding factual association was edited.

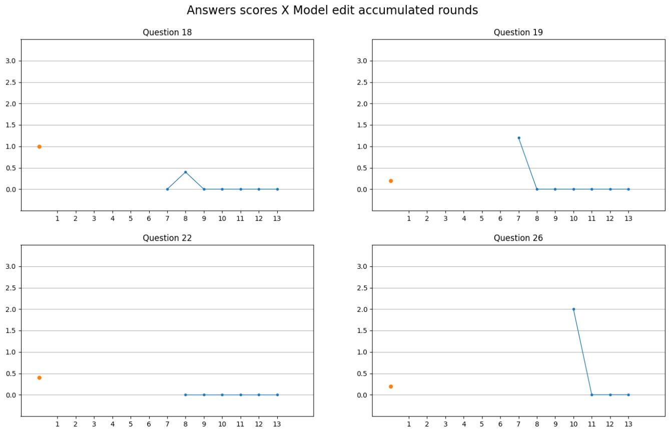

Fig. 5. Scores for answers to questions 18 (“What was Abhay Bhushan Pandey’s position at the Institute of Engineering and Rural Technology in Allahabad?”), 19 (“At which institute was Abhay Bhushan Pandey a Director?”), 22 (“What was Abhay Bhushan Pandey’s role in Xerox?”), and 26 (“What company did Abhay Bhushan Pandey co-found?”). Questions 18, 19, and 22 illustrate the fact the LLM was not able to achieve high scores to any question after half of the experiment, being question 26 the sole exception.

3.4 Comparing model editing performance against naive RAG

For comparing the edited LLM performance against the RAG approach the following baselines were considered:

1. Naive RAG backed by Llama 8B and Llama 70B LLMs from Groq AI infrastructure: those represent strong baselines for the experiment.
2. Naive RAG backed by the original Microsoft Phi 1.5 LLM, without any factual association edit: that is the direct comparison against the edited version.

The LLM editing poor performance on accumulated single factual associations already anticipate the results on the RAG comparison. As shown in figure 6, LLM editing approach were not able to successfully incorporate the knowledge of the Biography selected record, and therefore cannot compete with RAG approaches. While the strong baselines were able to provide good to perfect answers to all the questions, the Microsoft Phi 1.5 RAG baseline, despite have not being specifically trained for RAG, was still able to produce reasonable answers (score above 1.0) to 3 out of 9 questions, and could not answer only 2 of them.

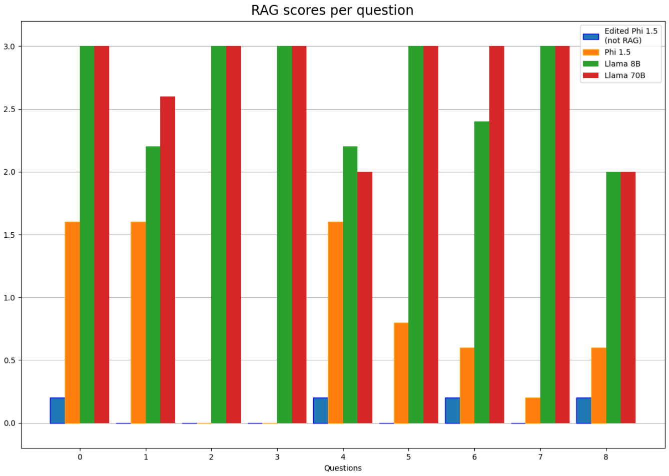

Fig. 6. Naive RAG baselines comparison against

4 Discussion and future works

The experiments could not insert all factual associations corresponding to a given Biography dataset record in a Microsoft Phi 1.5 LLM, making the edited

LLM successfully answer to questions related to that record, the same way a RAG system would be able to answer. Considering those results, the ROME LLM editing technique, to insert factual associations is not a RAG competitor.

However, the fact each LLM edit practically destroyed the knowledge inserted in the prior edits is a behavior not indicated in the ROME original work, and suggests some particularity of our research methodology. The poor performance in very few edits seems to be linked to the same subject of all edited factual associations: a quick experiment editing a factual association with a different subject indicated no impact on the previous editing.

The experiments also indicated the factual association editing is sensitive to the way the questions are posed against the incorporated fact: after editing the fact "Abhay Bhushan Pandey was a senior manager in Engineering and Development of Xerox", the edited LLM could not answer the two generated questions for the statement — "Who was a senior manager in Engineering and Development of Xerox?" and "What was Abhay Bhushan Pandey's role in Xerox?". However, the same edited LLM could reasonably answer a differently posed question: "Was Abhay Bhushan Pandey a senior manager of Xerox?".

Both behavior seems to be associated to the ROME technique approach of interpreting the MLP sub-layer as an associative memory for factual associations, linking a KEY (k) to a given VALUE (v). For a factual association to be edited in a LLM, the technique computes k* as the average activation after the non-linear transform of the MLP sub-layer of the **subject's** last token, and uses the prompt containing the **relation** and the **object** to optimize the last linear transform of that same MLP sub-layer produce the activations corresponding to the **object** to be edited. Hence, it seems the factual associations used for the experiments were all colliding, since their k* were the same. Also, questions which do not include both the **subject** and the corresponding **relation** might not properly make the LLM produce the desired **object**.

Considering the observations above, the following next experiments could correspond to future works on exploring the LLM editing capacity to achieve RAG-like answers:

1. Create factual association with the **subject** expanded to include part of the **relation** information, to avoid collision and enable multiple factual associations to a single entity.
2. Create additional factual associations relating the **object** to the **subject** in different ways, trying to expand the LLM generalization capacity to the giving fact.
3. Use the MEMIT (Mass-Editing Memory in a Transformer) technique [5], developed by the same research team of ROME, to include several factual associations edit in a LLM, as it spreads the edit across several LLM layers, as an attempt to minimize the edit impact on the existing knowledge.
4. Explore selecting specific samples when computing the uncentered covariance statistic for the MLP projection sub-layer weights, also as a way of minimizing the edit impact of the exiting knowledge, specially in a particular area or contexts.

Finally, all this research is publicly available; in particular, this Jupyter Python Notebook[6] contains most of the analysis reported on the document.

References

1. AI@Meta: Llama 3 model card (2024), https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md
2. Gu, J.C., Xu, H.X., Ma, J.Y., Lu, P., Ling, Z.H., Chang, K.W., Peng, N.: Model editing can hurt general abilities of large language models. arXiv preprint arXiv:2401.04700 (2024)
3. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems **33**, 9459–9474 (2020)
4. Meng, K., Bau, D., Andonian, A., Belinkov, Y.: Locating and editing factual associations in gpt. Advances in Neural Information Processing Systems **35**, 17359–17372 (2022)
5. Meng, K., Sharma, A.S., Andonian, A., Belinkov, Y., Bau, D.: Mass-editing memory in a transformer. arXiv preprint arXiv:2210.07229 (2022)
6. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[6] https://github.com/eduseiti/llm_editing_evaluation/blob/main/factual_associations/plot_3_step_results.ipynb