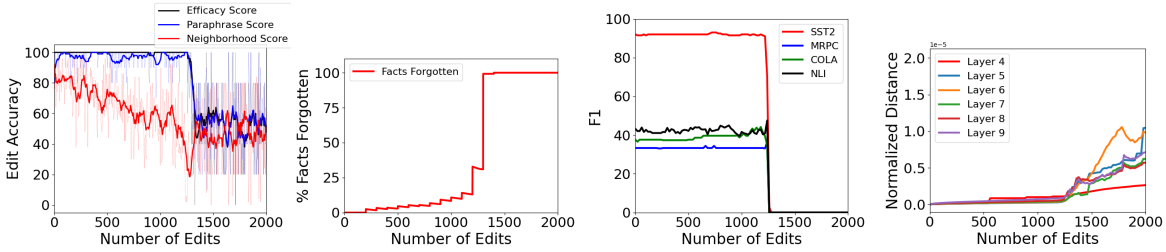

(a) Editing Proficiency (b) Forgetting (c) Downstream Performance (d) Distance

Figure 3: This figure shows the editing proficiency of MEMIT on GPT-J for Sample 1 over 2000 sequential edits made to the model.

call these edits ***disabling edits***, as they disable the model and make it unusable for downstream tasks. Note that these are facts that ROME is successfully able to edit in the model. Are these disabling edits a result of continuous sequential editing that accumulates over time or of one specific fact that is especially hard for a model to learn?

We find that when we edit the facts corresponding to disabling edits as the first edit made to the model, the model is still left completely disabled, and the normalized distance of the layer weights from the original weights is comparable to the distances seen around the spikes. This can be seen in Table 2, where we present the average normalized L2 norm between the edited model layer and its original weights when only one edit is made to the model. We find that disabling edits have three orders of magnitude larger distance than non-disabling edits. This shows that the disabling edits in ROME are not a result of continuous sequential editing of the model, but a fundamental limitation of ROME. We can describe disabling edits as facts that ROME is unable to successfully edit without crippling the model. Such disabling edits can also be a source of potential adversarial attacks.

4 Scaling MEMIT

In this section, we evaluate the performance of the Mass-Editing Memory in a Transformer (MEMIT) method (Meng et al., 2022b) when multiple sequential edits are made to the same model. We will follow the same procedure as followed in section 3, first evaluating the editing proficiency, fact forgetting and loss of performance on downstream tasks. We perform sequential editing on GPT-J (6B) using a random subset of 2000 examples from the CounterFact dataset when using MEMIT. This subset is a continuation of sample 1 in section 3.

Figure 3a shows the editing proficiency of MEMIT as a function of the number of edits made to the model. Note that here we edit one fact at a time for MEMIT. While MEMIT is able to make batched edits, we leave that analysis for future work. The dotted lines show a window size of 5 previous edits, whereas solid lines show a window size of 50 previous edits, same as in Figure 1c. We see that the efficacy score for MEMIT is not as high as ROME. **This means that knowledge edits made via MEMIT are not always successful, while in ROME we're always able to edit facts successfully**. We also see a continuous decline of neighborhood score for MEMIT as seen for ROME, showing that editing facts also start affecting other facts stored in the model.

Figure 3b shows the percentage of successfully edited facts that get forgotten as new facts are edited using MEMIT. We again begin to see two phases of forgetting - gradual and catastrophic. The catastrophic forgetting phase begins after approximately 1400 edits made to the model. When compared to ROME, we find that edits made using MEMIT have a much longer gradual forgetting phase across multiple samples. Additionally, we also find that **MEMIT forgets fewer previously edited facts when compared to ROME**, as seen in Figure 4, where MEMIT forgets almost three times fewer facts when compared to ROME. This can also be seen for other samples in appendix A.4.2.

Finally, the effect of the number of edits on the downstream performance of the model can be seen in Figure 3c. We see that the model maintains its ability to downstream tasks as more number of edits are made to the model. While this is true for GPT-J, we observe a gradual loss of performance with multiple edits for GPT2-XL (appendix A.3). In fact, we find that for GPT2-XL, the model loses the ability to do paraphrase detection long before the point of catastrophic forgetting, thus showcasing

Property	FT-C	ROME	MEMIT
EDITING EFFICACY	100%	100% until CF	< 100% until CF
EDIT LOCALITY	Very Low	High	Very High
AVERAGE DURATION BEFORE CF	CF not observed	Short	Long
DOWNSTREAM PERFORMANCE LOSS	High	High	Low
FACT FORGETTING PERCENTAGE	Very High	High	Low
SINGLE DISABLING EDIT	False	True	False

Table 3: Comparison between ROME and MEMIT at scale. CF refers to the point of catastrophic forgetting.

that the model can lose its ability of performing certain downstream tasks even before a disabling edit cripples the model.

Figure 3d shows the distance of the edited layers from the respective original layers. We find that the distance from the respective original layer increases gradually until approximately 1400 edits. After this, we find spikes in the distance between the edited layers and the original layers, which coincides with the points of catastrophic forgetting as seen in previous plots. We also evaluate if disabling edits are fundamental to MEMIT. We find that, if the fact that disables the model after a sequence of edits is edited first in the model, it does not lead to catastrophic forgetting. Thus, MEMIT is more robust to a single destabilizing edits when compared to ROME. We conjecture this is because a fact is stored within multiple layers of a model (Meng et al., 2022a,b), and editing the weights of a single layer to edit facts can lead to larger instabilities in the model. We summarize the properties of ROME and MEMIT at scale in Table 3, clearly showing MEMIT as a superior method across different parameters except editing efficacy.

5 Related Work

In this paper, we focus on model editing methods that modify the base language model's parameters. Some of these methods (De Cao et al., 2021; Mitchell et al., 2021) require training a hypernetwork (Chauhan et al., 2023) that generates new weights for the model being edited. Other methods (Meng et al., 2022a,b; Li et al., 2023a) directly update specific parts of the model after locating stored facts inside it. Gupta et al. (2024) unify these methods under the same framework called the preservation-memorization framework and enable batched editing with ROME, an algorithm they call EMMET. Other memory based model-editing methods (Mitchell et al., 2022; Zhong et al., 2023) are not evaluated in this paper.

While many of these methods have shown promise (Yao et al., 2023), recent work analyzing the after-effects of these editing methods have highlighted the shortcomings of these methods. Specifically, while some of these editing methods rank high on reliability, generalization and locality metrics (Yao et al., 2023; Mitchell et al., 2021, 2022; Meng et al., 2022a,b), the edited knowledge is not used consistently by the model. Cohen et al. (2023) propose a new evaluation system where the "ripple effects" or implications of an edited fact are evaluated. An example of such ripple effects would be - if an edited fact updates the president of a country to the new president, then prompting for the birthplace of the president should output the birthplace of the new president. Li et al. (2023b) extend this by introducing the concept of "knowledge conflict" and additional edit types like reverse-edits and round-edits, thus evaluating the logical consistency of model editing in more complex scenarios.

6 Conclusion

In this paper we analyze popular model editing techniques at scale. We use these methods to make multiple sequential edits to the same model and find that they fail in multiple ways. We find that ROME and MEMIT perform the best when scaled to multiple sequential edits as measured using metrics like fact forgetting and downstream performance. As we edit the models, we discover they undergo forgetting of previous knowledge and skills in two phases. Initially, the model gradually forgets previously edited facts and loses the ability to do downstream tasks, a phase which we call *gradual forgetting*. After that, the model abruptly loses all coherence and function including the ability to recall previously edited facts, perform downstream tasks and the ability to be edited, which is a realization of *catastrophic forgetting*. We also find that the source of these two phases of forgetting is that the layers being edited with these methods slowly drift away from their original weight values, thus becoming incompatible with the rest of the model.

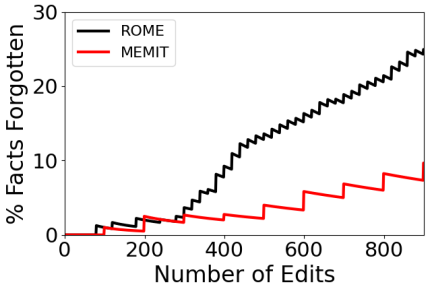

Figure 4: Compares the forgetting rate between ROME and MEMIT.

Practical use of model editing requires us to be able to make multiple sequential edits to a model. We find that these model editing methods, like other fine tuning techniques, are prone to catastrophic forgetting. To be able to scale such methods, we not only need to have high efficacy, specificity and generalization, but we also need these methods to preserve the model's existing abilities. With this paper, we call for an improved evaluation of model editing techniques at scale, including evaluating model performance on downstream tasks and ability to recall previously edited facts.

Finally, we want to stress upon the implications of the two phases of forgetting discovered in this paper for ROME and MEMIT. Gradual forgetting makes model editing techniques increasingly less effective as we sequentially edit facts, and hence limits their usefulness at scale. While catastrophic forgetting, which renders the model practically useless, caps the extent to which we can scale these methods. Thus we need to create model editing techniques that can counteract both gradual forgetting and catastrophic forgetting when scaled.

7 Limitations

The aim of our work is to present the efficacy of current model editing techniques at scale and the usefulness of our proposed evaluation framework when studying model editing techniques at scale. To do so, in this paper we study models of size 1.5 billion and 6 billion parameters, which are standard models used in previous works (Mitchell et al., 2021; Meng et al., 2022a,b). While we see consistent behavior of all model editing methods for the two sizes, it is possible that as models grow even larger, they respond differently to different model editing techniques. Additionally, some model editing methods like MEND (Mitchell et al., 2021) and MEMIT (Meng et al., 2022b) have the ability to perform batched edits, that is, make multiple edits is one gradient update. Effects of model editing techniques on larger model sizes, batch edits with increasing batch sizes, as well combining multiple batches of edits sequentially are not presented in this paper. We find that these aspects of model editing are a natural extension of our work but were out of scope for this paper due to space constraints. Yet these settings can be easily evaluated under the framework we have presented in this paper and have been left for future work.

References

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. *TAC*, 7:8.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Vinod Kumar Chauhan, Jiandong Zhou, Ping Lu, Soheila Molaei, and David A Clifton. 2023. A brief review of hypernetworks in deep learning. *arXiv preprint arXiv:2306.06955*.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2023. Evaluating the ripple effects of knowledge editing in language models. *arXiv preprint arXiv:2307.12976*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9.