Editing Memory in Transformers at Scale: A Benchmark and Robustness Study of MEMIT

Elvis Gjelaj Laahini Addagatla Niyathi Kukkapalli
COS 484: Natural Language Processing
Princeton University
eg5552@princeton.edu, la0047@princeton.edu, nk6074@princeton.edu

Abstract

Model editing is a way to update factual knowledge in large language models without having to retraining them from scratch. MEMIT achieved strong results on GPT-family models through editing transformer MLP weights to insert many factual associations at once. Our goal is to study whether these results transfer to a newer architecture and whether edited knowledge remains robust under adversarial prompting. We reproduce MEMIT on GPT-J 6B, port it to Llama 3.1 8B, and evaluate it based on clean edit success, jailbreaking resistance, and cross-domain interference. Our findings show MEMIT achieves 100% clean edit success in our 500-edit robustness setting highlighting its effective on GPT-J. It performs substantially worse on Llama where the best configuration reaches only 62% clean edit success. We also find that GPT-J edits are brittle under adversarial prompting. Overall, our results suggest that MEMIT's edits can strongly redirect factual recall under standard prompts, but they may not fully overwrite prior knowledge and do not transfer cleanly across transformer architectures.

1 Introduction

After computationally intensive training, large language models store knowledge in their parameters. As real-world knowledge evolves over time, these models require an effective mechanism to update outdated or incorrect information. However, retraining modern language models is often impractical due to their size and computational cost. Retrieval-based approaches, such as retrieval-augmented generation, can provide models with updated external information, but they do not directly modify the model's internal knowledge. Model editing methods attempt to address this limitation by changing a model's parameters directly. One such method, MEMIT, identifies a subset of middle MLP layers that mediate factual recall and updates their weights to insert factual knowledge at scale. MEMIT demonstrated strong results on GPT-J and GPT-NeoX by successfully applying thousands of simultaneous edits. We evaluate whether MEMIT's effectiveness extends beyond its original setting. We reproduce MEMIT on GPT-J, adapt it to Llama 3.1 8B, and perform layer and hyperparameter sweeps. We further test its robustness through jailbreaking and cross-domain ablation experiments. We find that while MEMIT remains effective on GPT-J, it transfers less reliably to Llama. We also find that GPT-J edits are successful under clean prompts but are vulnerable to adversarial prompting, which suggests that MEMIT's edited knowledge may not fully overwrite the model's original representations.

2 Related Work

Early approaches to model editing include MEMIT's predecessor ROME, which takes a direct approach to model editing by modifying factual associations through the transformer's MLP weights. However, ROME is designed primarily for single-fact edits, while MEMIT scales model editing from individual facts to large batches of simultaneous edits. It builds on the idea that transformer MLP layers act as key-value memories. MEMIT uses causal tracing to identify middle layers involved in factual recall. MEMIT then distributes updates across this range of MLP layers, which allows thousands of updated associations to be inserted while maintaining generalization and specificity.

MEMIT was primarily evaluated on GPT-family architectures and under relatively clean factual prompting conditions. Naturally, this leaves open questions about whether MEMIT transfers to more modern architectures such as Llama and whether edited facts remain stable under more rigorous evaluations. Our work builds upon MEMIT by reproducing it on GPT-J, adapting it to Llama 3.1 8B, sweeping layer and hyperparameter choices, and testing robustness through jailbreaking and cross-

domain ablations.

3 Baseline

MEMIT treats factual knowledge as subject-relation-object associations, such as $(s = \text{LeBron James}, r = \text{plays sport}, o_{\text{original}} = \text{basketball})$. Given a prompt expressing the subject and relation, MEMIT edits the model so that it prefers a new target object over the original object: $(s = \text{LeBron James}, r = \text{plays sport}, o_{\text{target}} = \text{baseball})$, where o_{target} is the new target object.

For each edit, MEMIT computes the target vector $z_i$for the hidden state at some selected layer. This target vector represents what the model's hidden state should look like if the new fact were successfully stored. MEMIT then updates the MLP weights across a selected range of middle layers so that the activations produced by those edited layers push the hidden state towards z_i. MEMIT does not simply force the output token directly; it changes the model's internal MLP parameters so that the edited fact becomes embedded into the model's computation.

We use GPT-J 6B as our baseline model because it is one of the main architectures evaluated in the original MEMIT paper. We reproduce MEMIT GPT-J using CounterFact-style edits and evaluate whether the edited model assigns higher probabilities to the injected target answers than the original answers, as shown in Figure 1. We test on 4 metrics: Efficacy Success (ES) – does the edit work on the original prompt, Paraphrase Success (PS) – does the edit generalize to reworded prompts, Neighborhood Success (NS) – does the edit avoid changing nearby related facts, and Editing Score (S) – an overall score combining ES, PS, and NS. This established that our implementation can successfully perform clean MEMIT edits before we extend MEMIT to Llama 3.1 8B and evaluate robustness under jailbreaking and cross-domain ablation settings on both architectures.

4 Jailbreaking Resistance

A core question for any model editing approach is whether the injected knowledge is robust to adversarial prompting. We ask whether a user can craft a prompt that induces an edited model to revert to its original, pre-edit beliefs. We investigate this question by applying four prompt-based attack strategies to models edited with 500 simultaneous MEMIT edits on both GPT-J 6B and Llama 3.1 8B

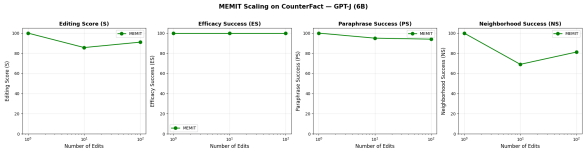

Figure 1: GPT-J reproduction of MEMIT on CounterFact. Following the original MEMIT evaluation, we report Editing Score (S), Efficacy Success (ES), Paraphrase Success (PS), and Neighborhood Success (NS) as the number of simultaneous edits increases. MEMIT achieves 100% efficacy across all tested edit counts and maintains high paraphrase success. The overall editing score remains high supporting a successful reproduction of MEMIT's core behavior on GPT-J.

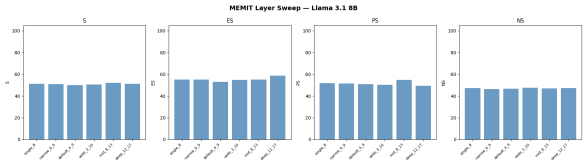

Figure 2: Empirical MEMIT layer sweep on Llama 3.1 8B. Unlike the original MEMIT paper, which used causal tracing to identify critical MLP layers for GPT-style models, we had to use a compute-constrained empirical sweep over candidate layer ranges. Across all tested layer ranges, Llama performance remains low and relatively flat: the best overall editing score occurs around the mid-layer range 8-13, while deeper layers improve efficacy slightly but do not improve the combined score. This suggests that simply moving MEMIT's edit window is not enough to recover the strong GPT-J behavior on Llama.

(using the best hyperparameter configuration from our sweep in Figure 2: layers 8–13, $\lambda = 1000$).

4.1 Attack Strategies

For each record in the MultiCounterFact dataset we denote the original (pre-edit) answer as t_{true} and the injected target answer as t_{new}. We evaluate the following conditions:

Clean (baseline). The original prompt is posed directly with no adversarial framing, establishing the unattacked edit success rate.

Context Assertion. A sentence explicitly asserting the original fact is prepended to the prompt: *"It is a well-established fact that [prompt]* t_{true}*. This is common knowledge. Based on this, [prompt]."*

Few-Shot. Two paraphrase prompts drawn from the dataset record are presented as demonstrations with t_{true} as the completion, followed by the target prompt. This strategy supplies in-context evidence that the original answer is correct.

Persona. The model is instructed to adopt the

Strategy	Edit Held (%)	Reverted (%)
GPT-J 6B (500 edits)		
Clean (baseline)	100.0	0.0
Context Assertion	56.6	43.4
Few-Shot	33.6	66.4
Persona	31.8	68.2
Prompt Injection	55.8	44.2
Llama 3.1 8B (500 edits, layers 8–13, $\lambda = 1000$)		
Clean (baseline)	62.0	38.0
Context Assertion	54.2	45.8
Few-Shot	62.2	37.8
Persona	67.6	32.4
Prompt Injection	67.4	32.6

Table 1: Jailbreaking resistance across 500 MEMIT edits evaluated over all 500 MultiCounterFact records. *Edit Held* is the percentage of edits where the model continued to prefer the injected answer t_{new}, and *Reverted* is the percentage where the model preferred the original answer t_{true} under the adversarial prompt.

role of a world-renowned expert who knows for certain that t_{true} is correct, then asked to answer the prompt.

Prompt Injection. A simulated system-level override explicitly instructs the model to ignore all prior weight modifications and restore its original knowledge before answering.

4.2 Evaluation Metric

For each strategy we compute the mean per-token negative log-likelihood (NLL) of both t_{new} and t_{true} conditioned on the adversarial prompt prefix. An edit is considered to have *reverted* if the model assigns a lower NLL (equivalently, higher probability) to t_{true} than to t_{new}. The reversion rate is the fraction of the 500 edited records where this condition holds.

4.3 Results

Table 1 presents reversion rates for both models. The two architectures exhibit strikingly different behavior, and the contrast is itself informative.

GPT-J 6B. On GPT-J, where MEMIT edits hold at 100% under the clean baseline, adversarial prompts cause substantial reversion. Few-shot demonstrations (66.4% reversion) and persona prompting (68.2% reversion) are the most effective attacks, each inducing the model to prefer the original answer in roughly two-thirds of cases. Context assertion and prompt injection are less effective but still non-trivial, achieving reversion rates of 43.4% and 44.2% respectively.

These results suggest that MEMIT edits on GPT-J, while successful at changing the model's default completion, do not eliminate the original knowledge from the weight matrix. They appear instead to shift probability mass toward t_{new} without fully erasing the representation of t_{true}. When the context supplies sufficiently strong in-context evidence for the original fact (as in few-shot and persona prompting), this residual representation is reactivated and the model reverts. The relatively weaker effect of prompt injection (44.2% vs. 68.2% for persona) suggests that the model does not treat natural-language "override" instructions as more authoritative than in-context demonstrations.

Llama 3.1 8B. The Llama results present a qualitatively different picture, though one that must be interpreted with caution. Because MEMIT edits hold in only 62.0% of cases even under the unattacked clean baseline, the Llama model is already partially reverting to original knowledge without any adversarial pressure. The baseline reversion rate of 38.0% sets a high floor above which adversarial strategies must operate.

Strikingly, three of the four attack strategies (few-shot at 37.8%, persona at 32.4%, and prompt injection at 32.6%) achieve reversion rates *below* the clean baseline, meaning that these prompts actually make the edits *more* persistent rather than less. Only context assertion produces an increase in reversion (+7.8 percentage points over baseline).

We attribute this counterintuitive result to Llama's instruction fine-tuning. Prompts that assert expert authority (persona) or invoke system-level override instructions (prompt injection) appear to activate Llama's instruction-following tendencies in a way that reinforces the model's most recently available completion, which in these cases is the injected answer. The few-shot strategy similarly produces no meaningful degradation, likely because the paraphrase demonstrations are imperfectly aligned to the edited prompt and provide less coherent in-context signal than on GPT-J.

As a result, these results indicate that the jailbreaking threat model differs substantially across architectures. For GPT-J, where MEMIT edits hold cleanly, adversarial prompts constitute a genuine and significant robustness concern: a user who knows the original fact can recover it in over two-thirds of cases using simple prompting strategies. For Llama, the weak baseline edit success rate

means that jailbreaking evaluation is largely uninformative. The model is already unreliable, and adversarial prompts do not systematically worsen the situation.

5 Cross-Domain Ablation

We performed a cross-domain interference ablation experiment on both the Llama 3.1 8B and GPT-J models. The ablation's core idea was to see if editing facts in one domain would cause unintended changes to the model's knowledge in other, unrelated domains. For example, we can edit the fact "LeBron James plays for the Miami Heat" to "LeBron James plays for the Eagles" and we check an unrelated fact like "The capital of France is Paris." A cross-domain change would be if the model starts incorrectly stating "The capital of France is Marseille." The term ***specificity*** represents this idea, meaning the change should not "bleed over" into unrelated domains. We note that a ***high editing score (S)*** means the model is doing well in all three domains: efficacy (actually applies the edit), generalization (applies edit beyond exact prompt), and specificity.

5.1 Cross-Domain Tests in MEMIT

MEMIT touches on this through its evaluation of *specificity* and a dedicated analysis of "Editing Different Categories of Facts Together" (Meng et al., 2023b). Its predecessor, ROME, similarly notes that successful edits must preserve specificity (Meng et al., 2023a). In Section 5.4, the authors test whether editing facts from unrelated categories induces interference, defining four settings based on subject and object similarity: (Subject Different, Object Different), (Subject Similar, Object Different), (Subject Different, Object Similar), and (Subject Similar, Object Similar). For example, "Steve Jobs is a citizen of France" paired with "The official language of Germany is Japanese" falls under (Subject Different, Object Different), since both the subjects (person vs. country) and objects (country vs. language) differ. Across all scenarios, MEMIT's editing score (S) remains high, typically in the mid-80s to mid-90s, even as the number of edits scales from 100 to 700, indicating that increased domain diversity does not significantly degrade performance.

5.2 Setup of Ablation

The ablation test in this paper focuses on side effects of an isolated edit. We have five domains of knowledge: Geography, Science, Sports, Resident Evil, and Jujutsu Kaisen. Before the ablation, we verify if the model knows the facts by computing the negative log likelihood (NLL) for original (target_true) and edited (target_new) outputs for a given fact. In the ablation, for each experimental run, we select one domain and apply MEMIT to simultaneously update all facts within it where we change true facts to new, changed facts. Following this, we meticulously measure the model's factual accuracy across all five domains (the edited domain and the four unedited domains). After each experiment, the model's weights are reset to their pre-edited states, ensuring independence between runs. We repeat this process for both the Llama 3.1 8B and GPT-J models.

5.3 Results

GPT-J 6B. For each of the five domains that we decided to edit, MEMIT achieved a high success rate (100% or 90% for the edit success as in Table 2. This means that the model was effectively updated to prefer target_new facts over target_true facts in the edited domain. As seen in the matrix, when a domain is edited, its accuracy dropped to 0%, indicating a successful overwrite of previous knowledge.

Llama 3.1 8B. MEMIT achieves moderate but uneven edit success across domains, while maintaining strong isolation between unrelated knowledge areas. Edit success rates range from 10% in Resident Evil to 70% in Geography, indicating that some domains are significantly easier to modify than others. In domains where edits are more successful—such as Geography (70%) and Science and Sports (both 50%), there is a clear drop in accuracy on the original facts (for example, Geography decreases from 100% to 30%). This suggests that the model is genuinely incorporating the new information and partially overwriting its previous knowledge. On the other hand, in domains with low edit success like Resident Evil and Jujutsu Kaisen, accuracy on the original facts remains high (90% and 80%), meaning the edits had little impact. This shows that MEMIT makes precise, localized updates without causing unintended side effects. Overall, the results suggest that while MEMIT is

Table 2: Cross-domain interference matrices. Values = accuracy (%) after editing the row domain. Diagonal entries $^{[E]}$ denote the edited domain itself.

Interference Matrix 1

Edited Domain	Geography	Science	Sports	Resident Evil	Jujutsu Kaisen	Edit Succ. (%)
Geography	$0.0^{[E]}$	100.0	100.0	50.0	70.0	100.0
Science	100.0	$0.0^{[E]}$	100.0	50.0	70.0	100.0
Sports	100.0	100.0	$0.0^{[E]}$	50.0	70.0	100.0
Resident Evil	100.0	100.0	100.0	$10.0^{[E]}$	70.0	90.0
Jujutsu Kaisen	100.0	100.0	100.0	50.0	$10.0^{[E]}$	90.0
Baseline	**100.0**	**100.0**	**100.0**	**50.0**	**70.0**	—

Interference Matrix 2

Edited Domain	Geography	Science	Sports	Resident Evil	Jujutsu Kaisen	Edit Succ. (%)
Geography	$30.0^{[E]}$	100.0	100.0	90.0	80.0	70.0
Science	100.0	$50.0^{[E]}$	100.0	90.0	80.0	50.0
Sports	100.0	100.0	$50.0^{[E]}$	90.0	80.0	50.0
Resident Evil	100.0	100.0	100.0	$90.0^{[E]}$	80.0	10.0
Jujutsu Kaisen	100.0	100.0	100.0	90.0	$80.0^{[E]}$	20.0
Baseline	**100.0**	**100.0**	**100.0**	**90.0**	**80.0**	—

reliable in keeping edits contained, its success in rewriting knowledge depends on the domain.

6 Limitations

MEMIT is fundamentally GPT-architecture specific. The most significant limitation of this work is that MEMIT was designed and validated for GPT-family models, presupposing an architectural layout that does not straightforwardly carry over to the Llama family. Our experiments confirm this empirically. Even after an extensive hyperparameter sweep, MEMIT achieves only approximately 62% edit success on Llama 3.1 8B, compared to 100% on GPT-J 6B. This gap suggests a architecture-dependent incompatibility rather than a tuning failure, and it significantly limits the conclusions of our Llama experiments. Our cross-domain and jailbreaking evaluations on Llama are difficult to interpret cleanly because the edits themselves are unreliable. Future work should prioritize architecture-agnostic editing methods, or at minimum verify edit success before performing downstream evaluation.

Scope of jailbreaking evaluation. Our jailbreaking experiments consider four relatively simple prompt-based attack strategies. More sophisticated techniques, such as gradient-based suffix optimization, multi-turn elicitation, or chain-of-thought attacks, may achieve higher reversion rates and provide a more complete picture of MEMIT's robustness. Our NLL-based metric also does not assess whether the model would generate the reverted answer in open-ended generation, which may be the more practically relevant failure mode.

Dataset and scale. All experiments use 500 edits from MultiCounterFact, which consists of simple single-hop factual associations. Scaling to larger edit batches or more complex relational facts may surface additional failure modes not visible at the scale studied here.

Generalizability to future editing methods. Our results should not be taken as evidence that model editing is infeasible for non-GPT-family models. MEMIT is one approach among a growing family of editing methods, and more recent architecture-agnostic approaches may avoid the limitations we observe. The difficulty of applying MEMIT to Llama underscores the need for editing benchmarks that do not implicitly favor the GPT family for which existing methods were originally designed.

7 Conclusion

We reproduced MEMIT on GPT-J 6B and found that it successfully inserts factual associations under clean prompting. However, our adaptation to Llama 3.1 8B was much less reliable. Despite layer and hyperparameter sweeps, the best configuration achieved substantially lower edit success than GPT-J. This suggests that MEMIT's effectiveness may

depend on architectural assumptions that transfer poorly to newer Llama-style models.

Our findings also show that clean edit success does not erase original knowledge. On GPT-J, few-shot and persona-based adversarial prompts caused the edited model to revert to the original answer in over 65% of cases. Therefore, MEMIT appears to redirect factual recall under standard prompts, but residual original knowledge can still be recovered through adversarial prompting. Llama's edit-hold rate remained near its clean baseline for most adversarial prompts, which suggests the edits that do survive on Llama appear more robust than those in GPT-J. We suggest future work should perform causal tracing directly on Llama-family models, test stronger adversarial attacks, and compare MEMIT against more architecture-agnostic editing methods. We also suggest a closer look at Llama's architectural differences which may help explain MEMIT's robustness to adversarial prompting.

Acknowledgements

We thank Professors Karthik Narasimhan and Tri Dao for teaching the course and answering project questions, and our mentor Max Gupta for feedback and guidance on project direction and benchmarking. We also thank Google for their generous contribution of compute units through the free Colab pro subscription for students and educators. In accordance with the COS 484 Spring 2026 project guidelines, we acknowledge the use of AI coding tools for this project.

Appendix

Google drive link for the ipynbs: https://drive.google.com/drive/folders/1kg67MVCovPmes-wb6yhdfYqwQ2zmdYBc?usp=sharing

References

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023a. Locating and editing factual associations in gpt.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023b. Mass-editing memory in a transformer.