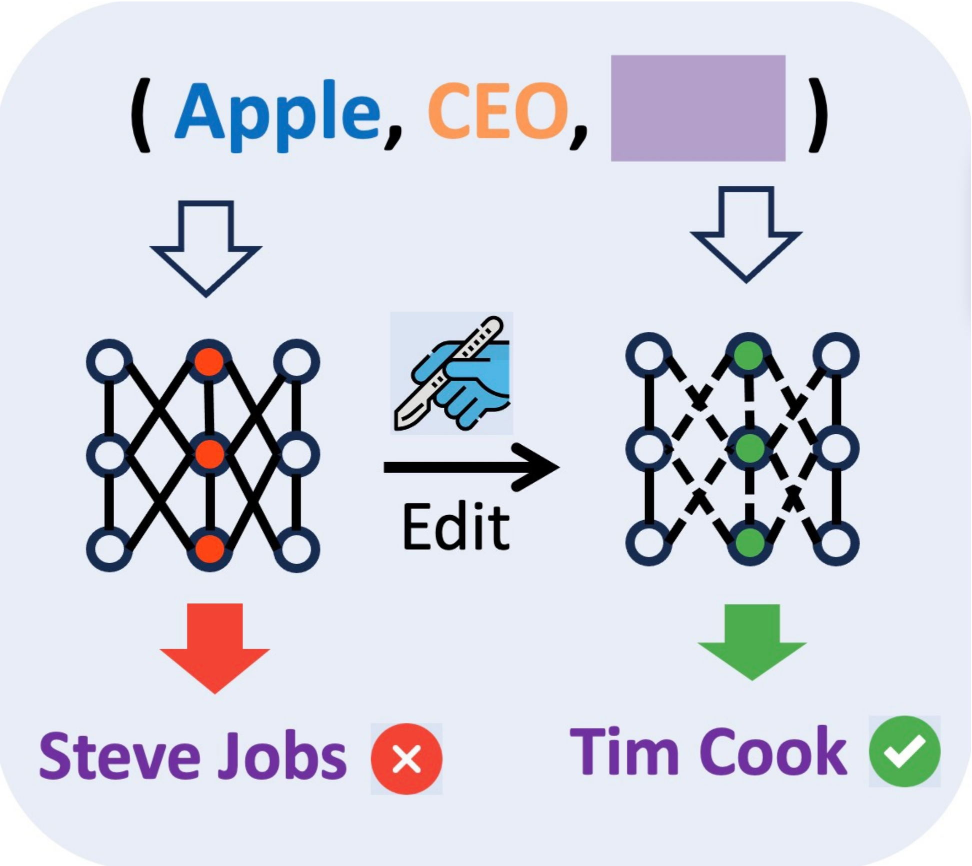
Navigating the Dual Facets: A Comprehensive Evaluation of Sequential Memory Editing in Large Language Models

Zihao Lin, Mohammad Beigi, Lifu Huang
April 5th, 2024, Virginia Tech

What is Memory Editing?

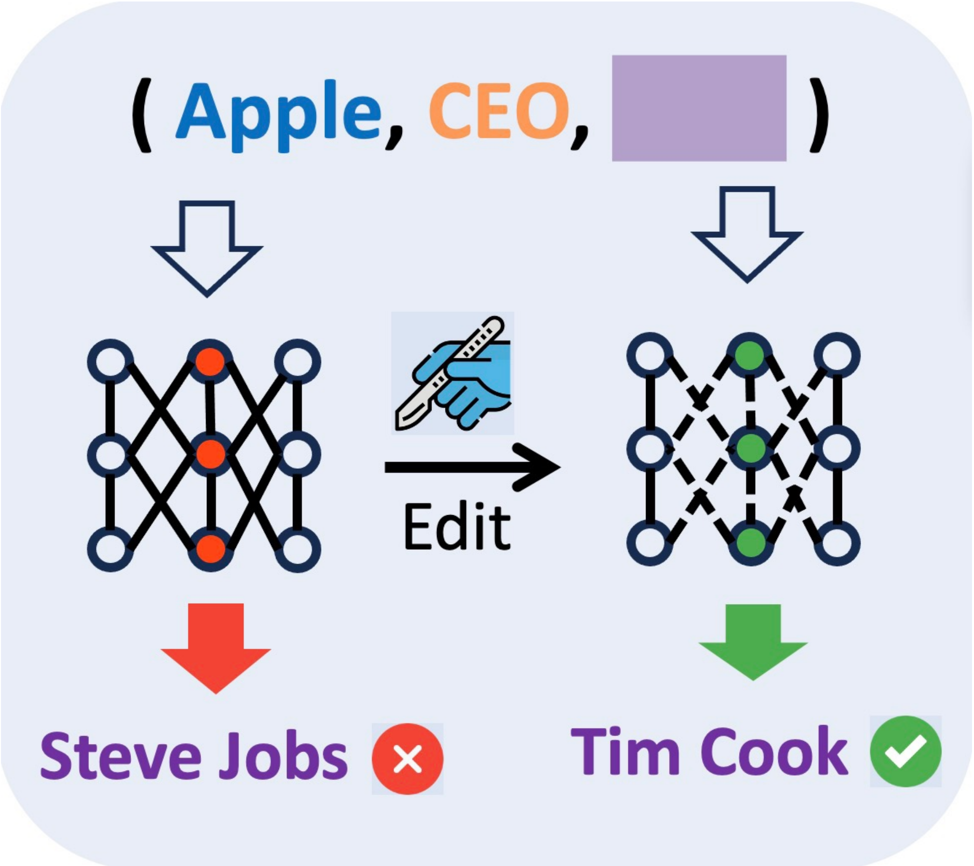
- Memory Editing (ME) was introduced as an effective method to correct erroneous facts or inject new knowledge into Large Language Models (LLMs) without changing unrelated knowledge.



What is Memory Editing?

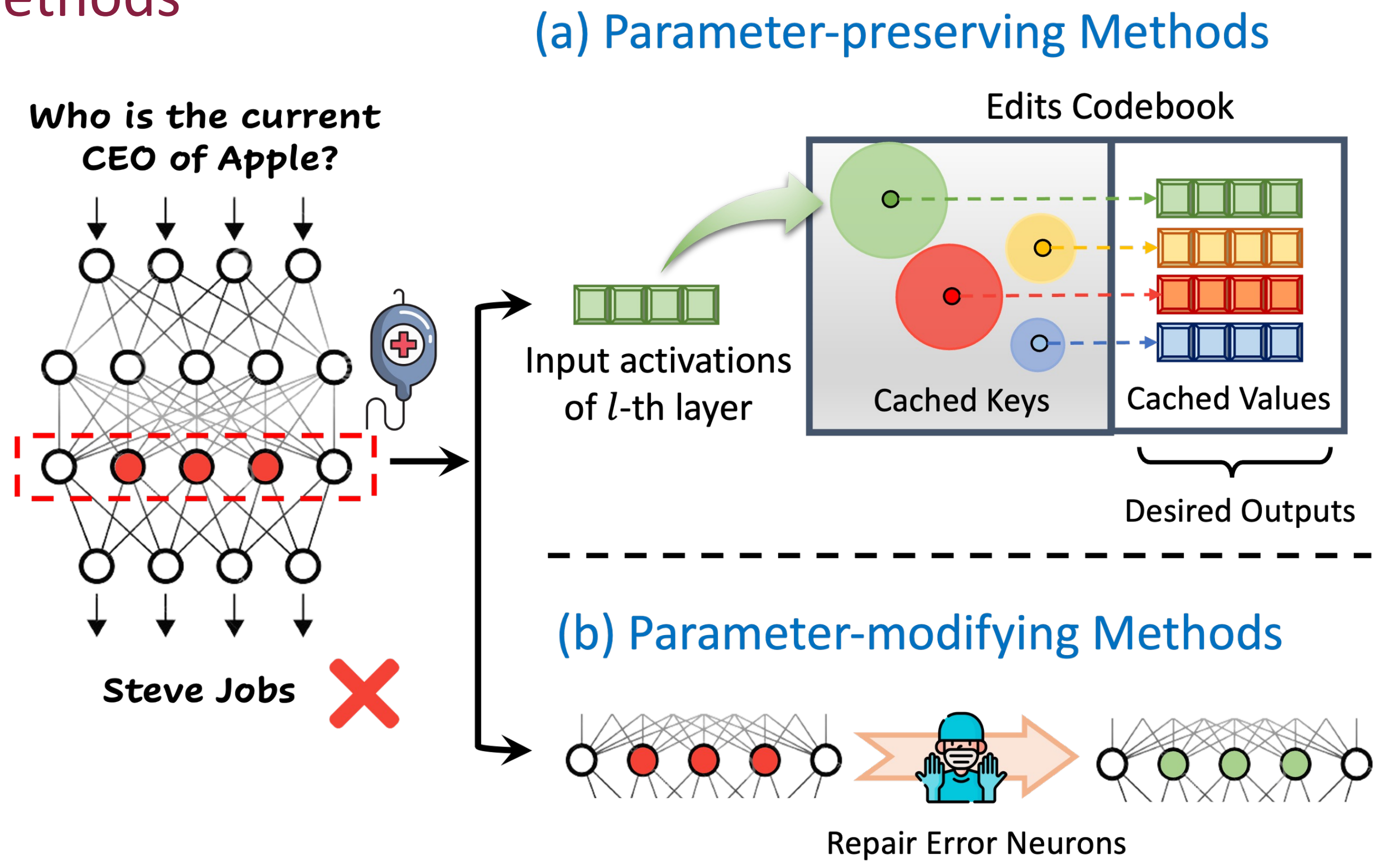
- Memory Editing (ME) was introduced as an effective method to correct erroneous facts or inject new knowledge into Large Language Models (LLMs) without changing unrelated knowledge.

- ME vs. Finetune:
  - ME does not change all the parameters of LLM.
  - ME is GPU & time efficient.



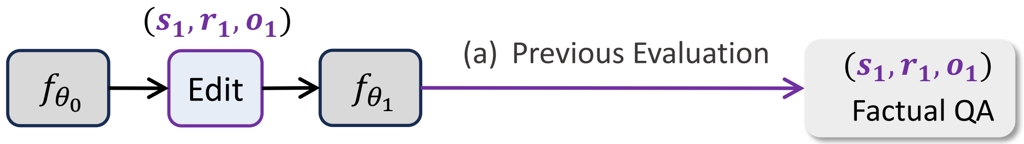
Types of Memory Editing Methods

- Two categories of ME methods:
  - parameter-modifying ME methods
  - parameter-preserving ME methods



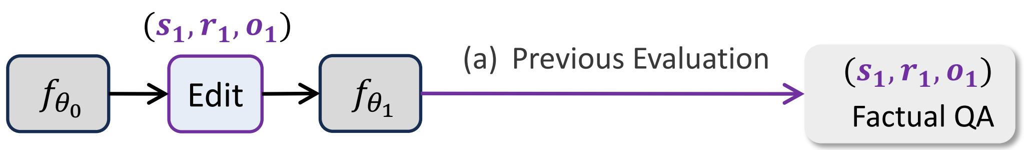
Motivation

- Previous studies evaluating and analyzing ME methods have two critical limitations:
  - They only consider the performance of LLMs after every single editing.
  - They only concentrate on assessing ME's impact on factual knowledge (s, r, o).

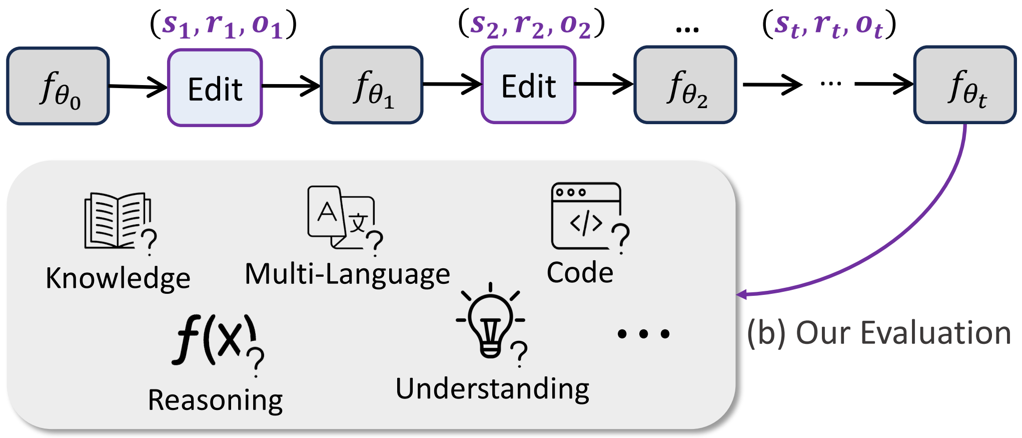


Motivation

- Previous studies evaluating and analyzing ME methods have two critical limitations:
  - They only consider the performance of LLMs after every single editing.
  - They only concentrate on assessing ME's impact on factual knowledge (s, r, o).



- To address these limitations, our study comprehensively evaluates the general capabilities of memory-edited LLMs in sequential editing scenarios.



General Capabilities of LLMs

- Eight evaluation datasets across six main capabilities of LLMs:
  - Professional Knowledge: MMLU [1]
  - Common Sense Knowledge: CommonsenseQA [2]
  - Logical Reasoning: MATH [3], BBH [4], SuperGLUE-AX-b [5]
  - Reading Understanding: C3 [6]
  - Multilingual Proficiency: TyDiQA [7]
  - Code Generation: MBPP [8]

Memory Editing Methods

- Four memory editing methods across two categories:
  - Parameter-modifying ME methods:
    - MEND [9]
    - ROME [10]
    - MEMIT [11]
  - Parameter-preserving ME method:
    - GRACE [12]

Experiments Settings

- Large Language Models:
  - LLaMA-2-7b [13]
  - LLaMA-2-7b-Chat [13]
  - LLaMA-2-13b [13]
- Editing Dataset:
  - Randomly select 100 samples from the ZsRE [14] as the editing dataset.

Evaluation Results of Memory Editing

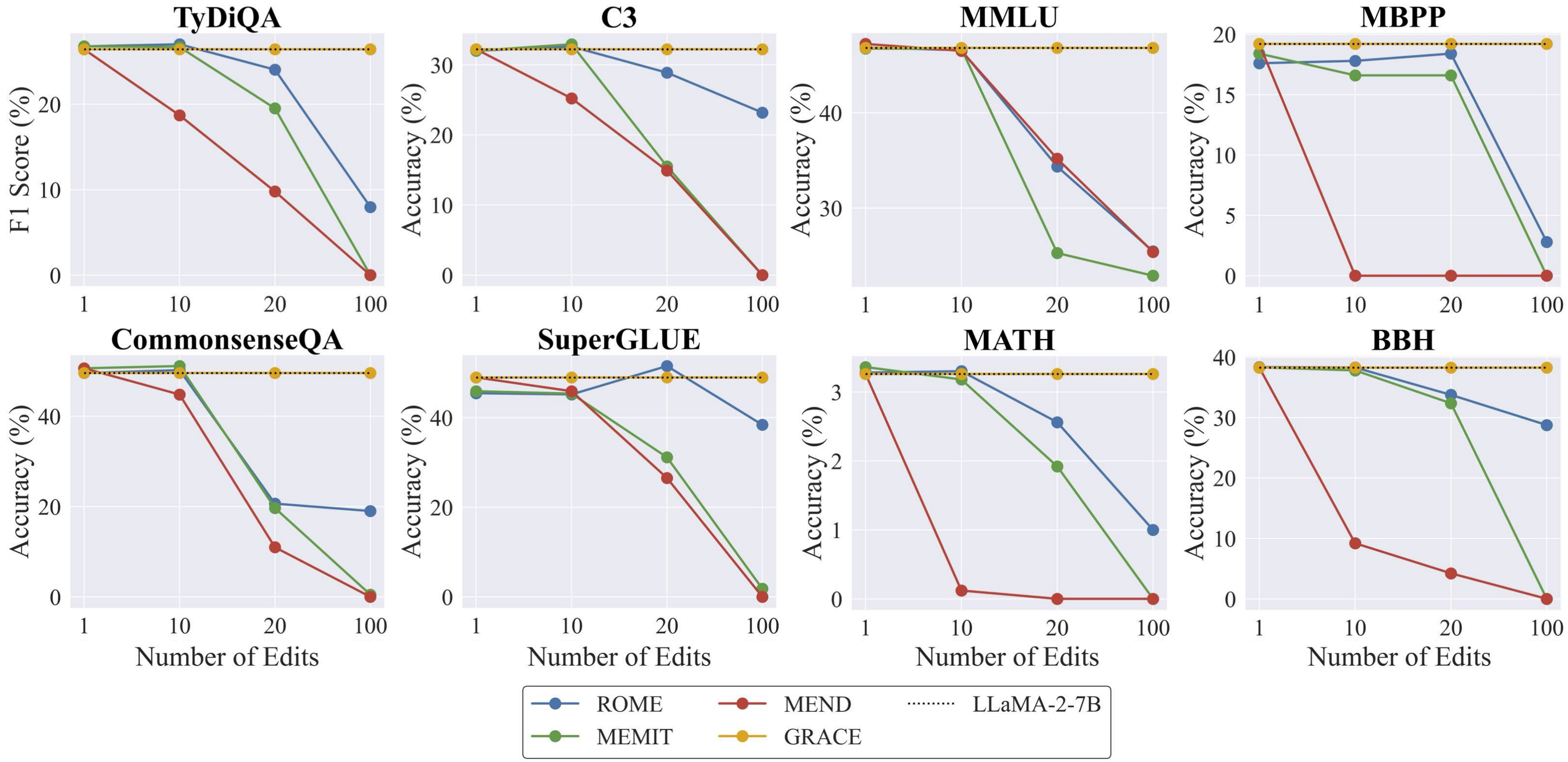
Evaluation of Downstream Tasks

- Evaluate Llama-2-7b on eight downstream tasks.
- Sequentially edit 1, 10, 20, 100 times.

Evaluation of Downstream Tasks

- Evaluate Llama-2-7b on eight downstream tasks.
- Sequentially edit 1, 10, 20, 100 times.

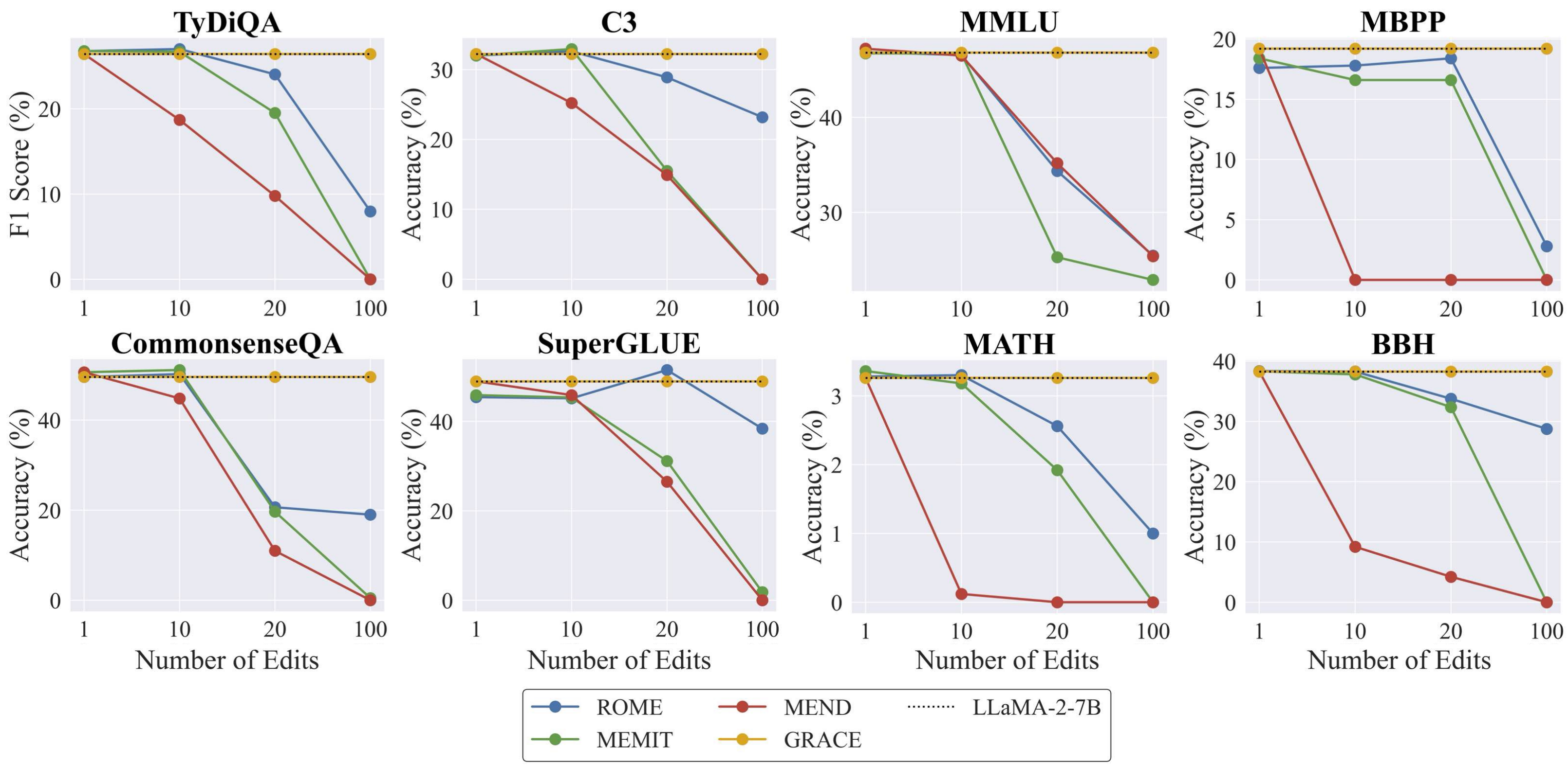
- Modifying-parameter ME methods systematically hurt the general capabilities of LLM after sequential editing.



Evaluation of Downstream Tasks

- Evaluate Llama-2-7b on eight downstream tasks.
- Sequentially edit 1, 10, 20, 100 times.

- Parameter-preserving ME method, GRACE, maintains the broad capabilities of LLMs.



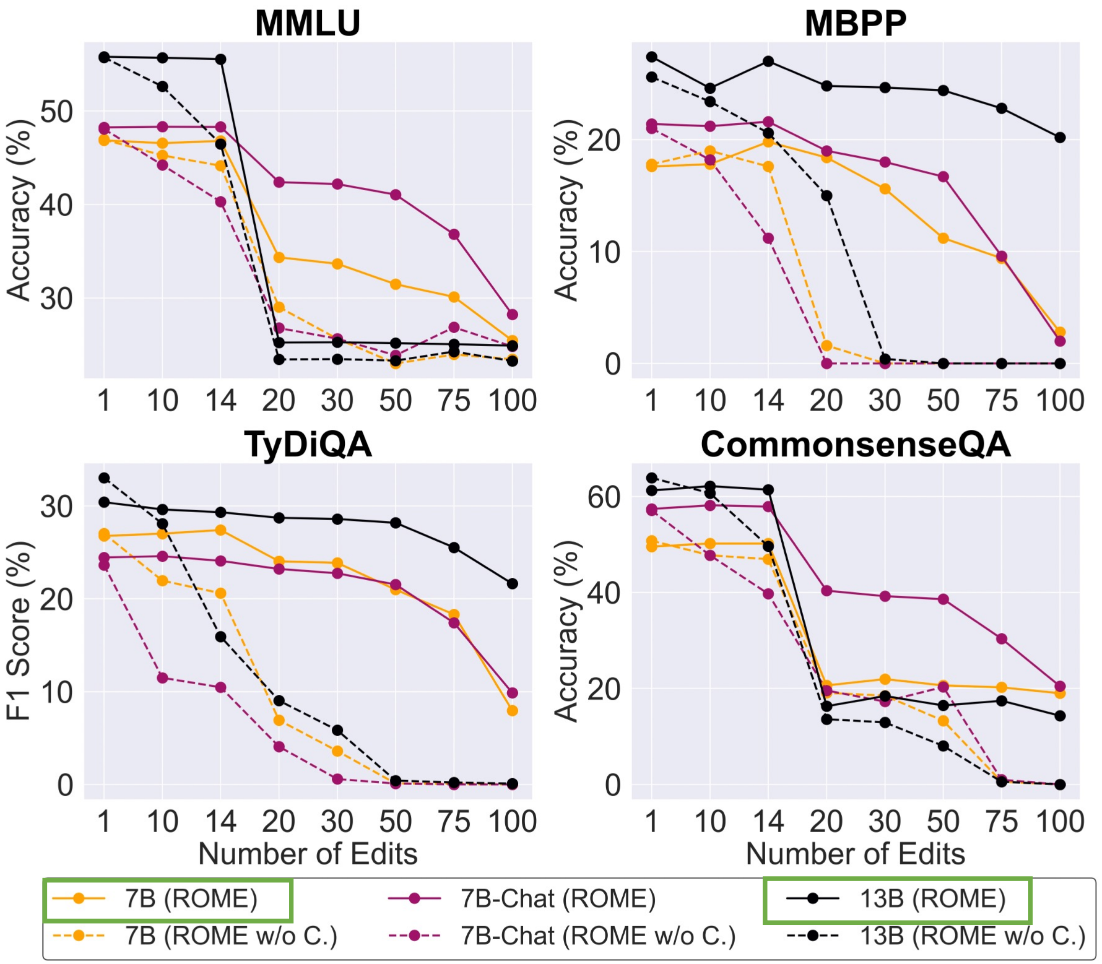
Effect of Model Checkpoints and Sizes

- Evaluation Datasets:
  - MMLU: High school/College Examination
  - CommonsenseQA: Common Sense Question Answering
  - MBPP: Code Generation
  - TyDiQA: Multi-language Understanding

Effect of Model Checkpoints and Sizes

- Evaluation Datasets:
  - MMLU: High school/College Examination
  - CommonsenseQA: Common Sense Question Answering
  - MBPP: Code Generation
  - TyDiQA: Multi-language Understanding

- Increasing models' parameters is beneficial to MBPP and TyDiQA.

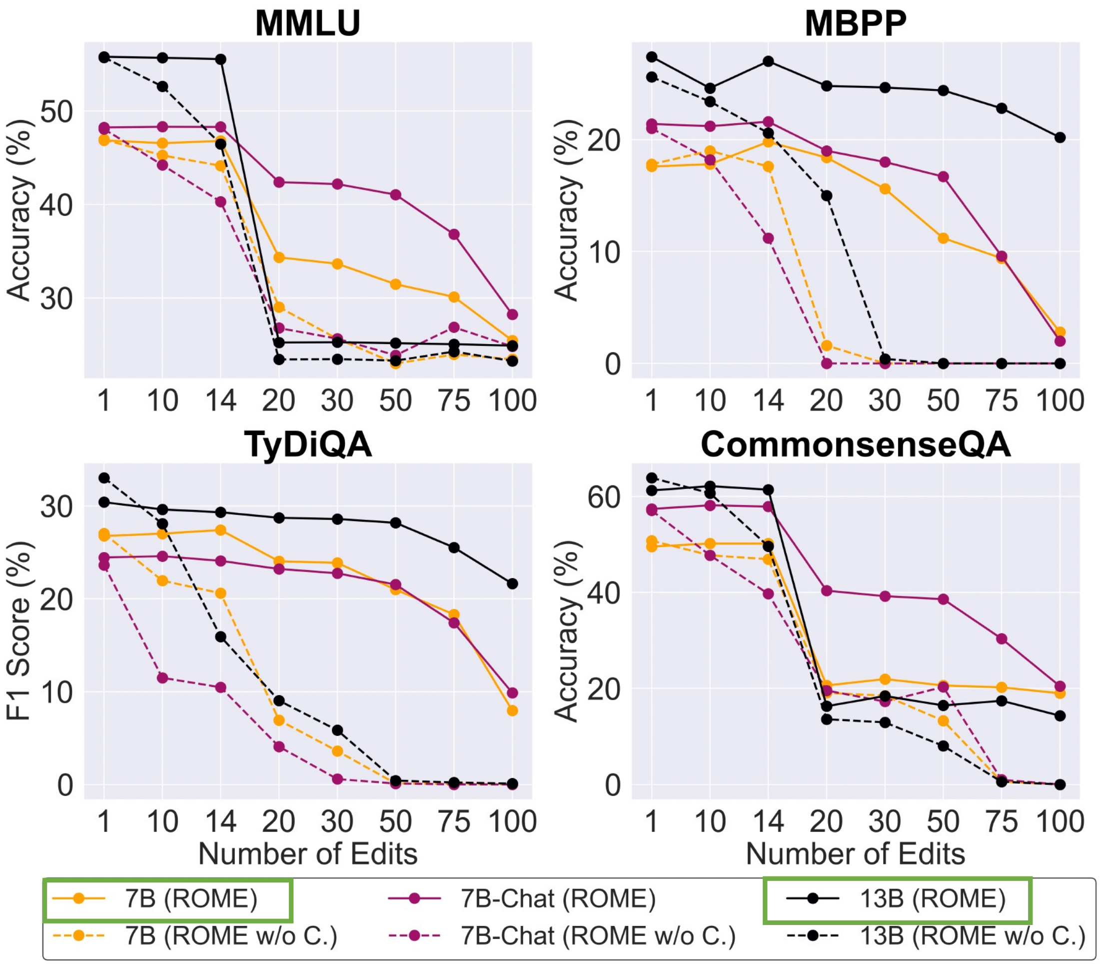


VIRGINIA TECH

Effect of Model Checkpoints and Sizes

- Evaluation Datasets:
  - MMLU: High school/College Examination
  - CommonsenseQA: Common Sense Question Answering
  - MBPP: [Code Generation]
  - TyDiQA: [Multi-language Understanding]

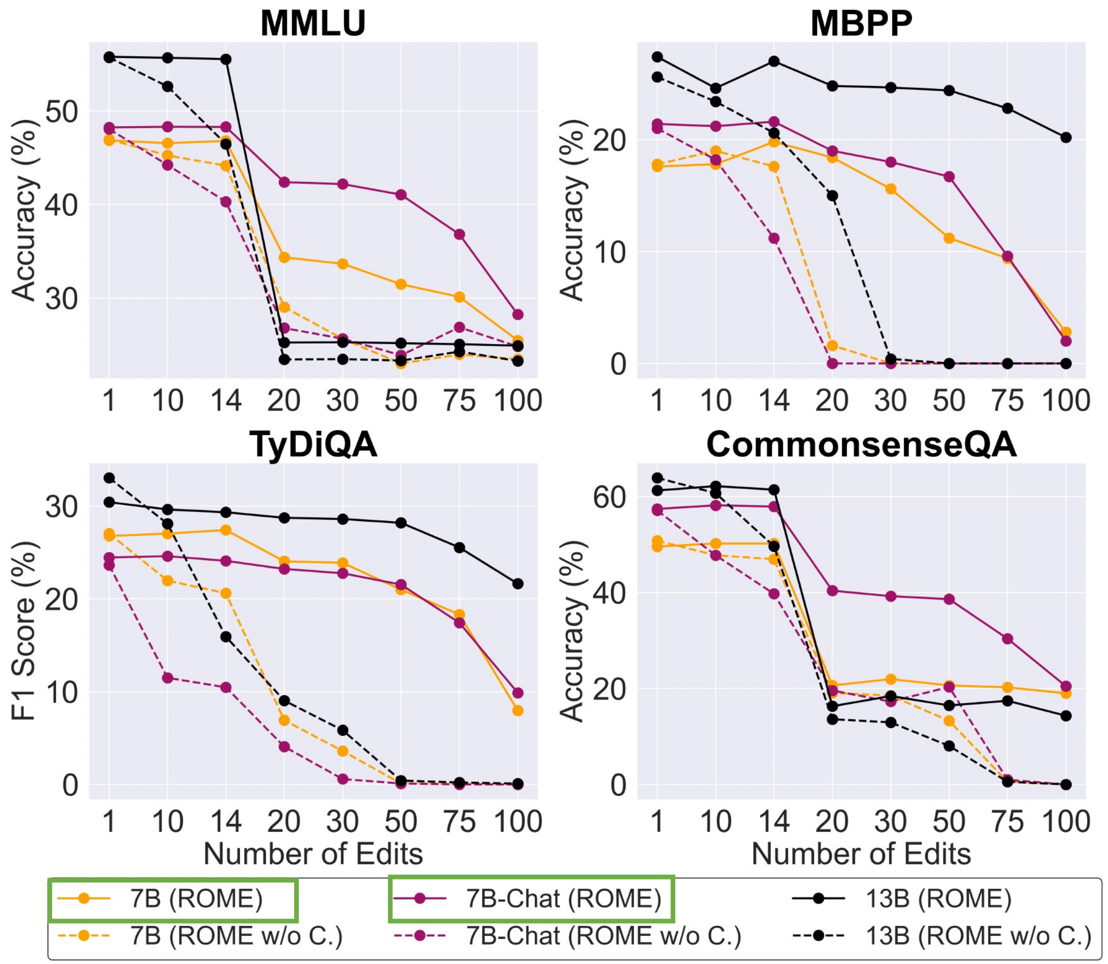
- Hypothesis: more parameters mean that there are enough parameters to store different knowledge in different parameters, which reduces the negative influence.



VIRGINIA TECH

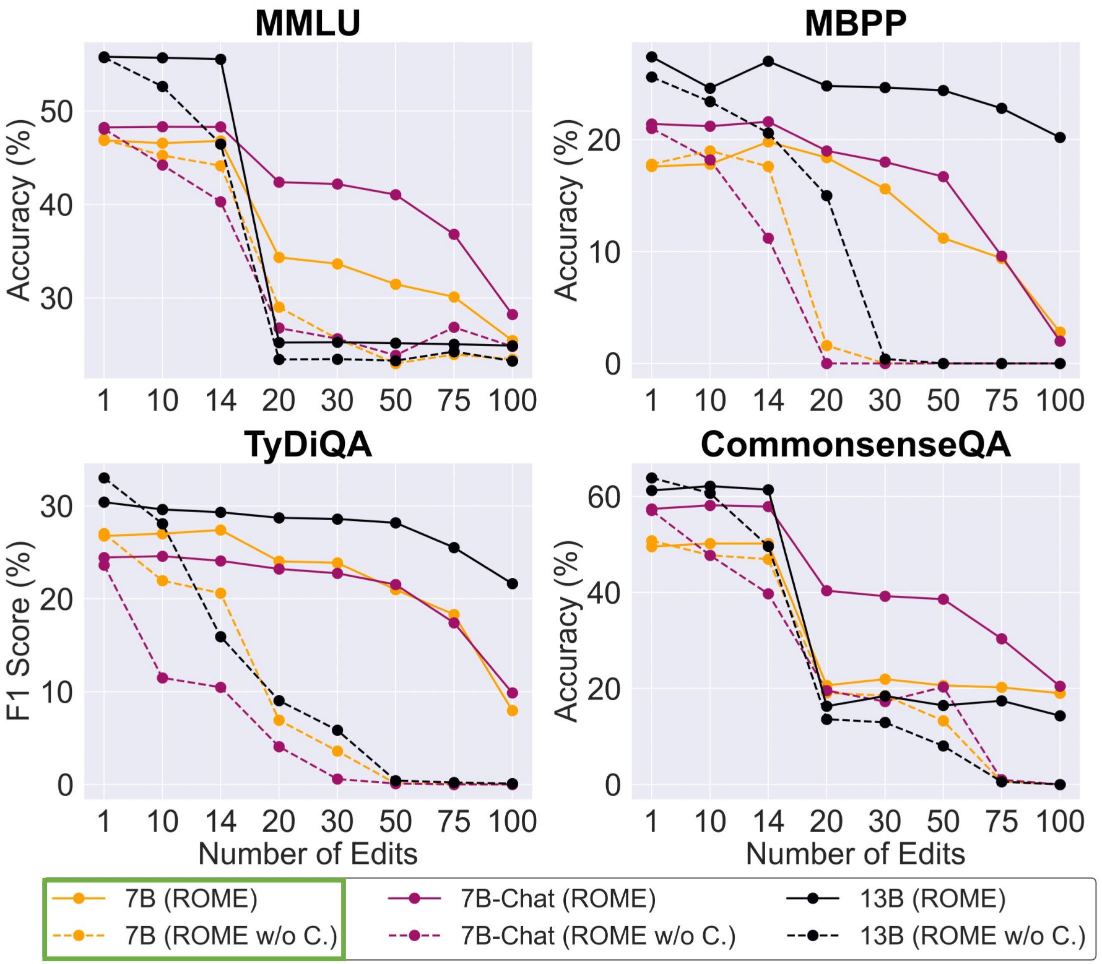
Instruction Tuning and Its Implications

- After instruction tuning, the model has more robustness on MMLU and CommonsenseQA, whose inputs are also similar to dialogue in English.



Constraint Methods in ROME

- ROME utilizes 100,000 Wiki Knowledge, which is unrelated to edited knowledge, and applies a constraint method to avoid the edited LLM forgetting some unrelated knowledge.

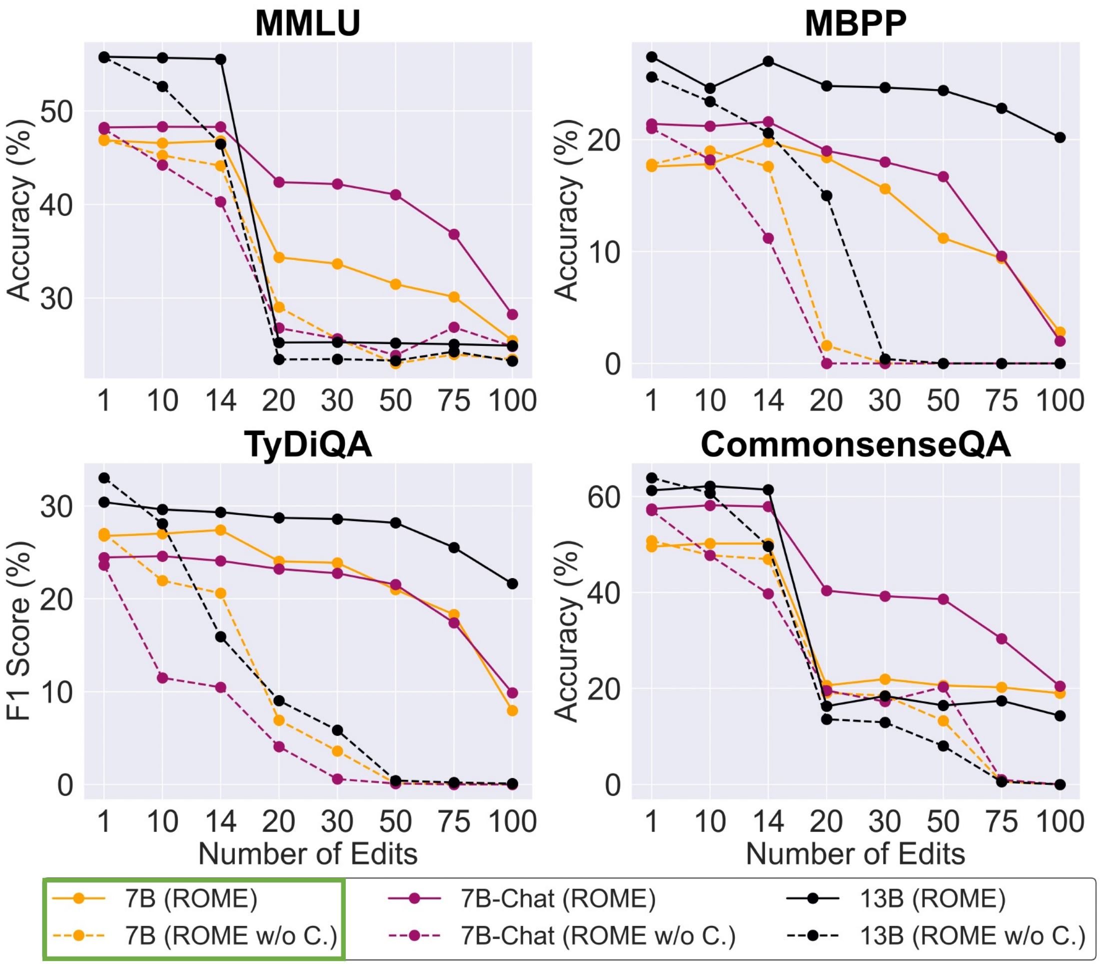


VIRGINIA TECH

Constraint Methods in ROME

- ROME utilizes 100,000 Wiki Knowledge, which is unrelated to edited knowledge, and applies a constraint method to avoid the edited LLM forgetting some unrelated knowledge.

- Adding constraints is beneficial to maintain general capabilities during sequential editing but cannot fully avoid such damage.



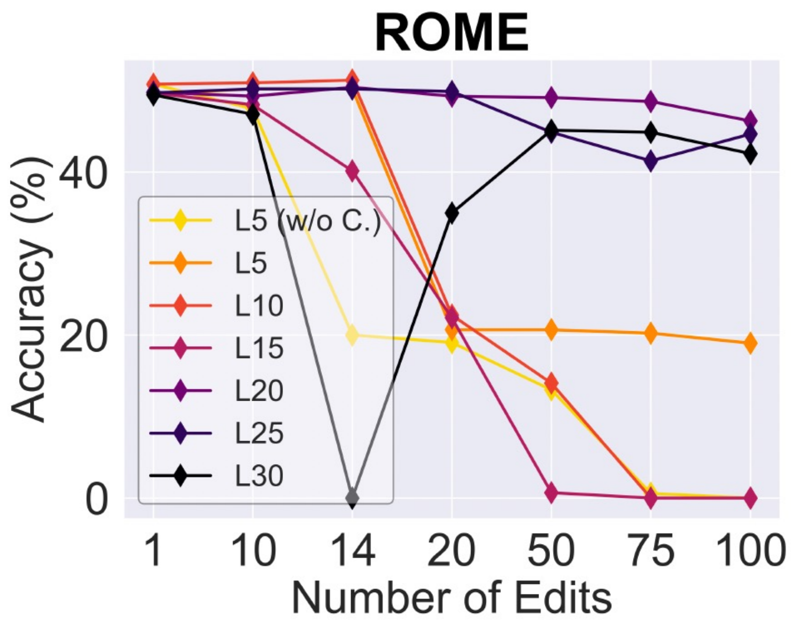
Layers to Edit

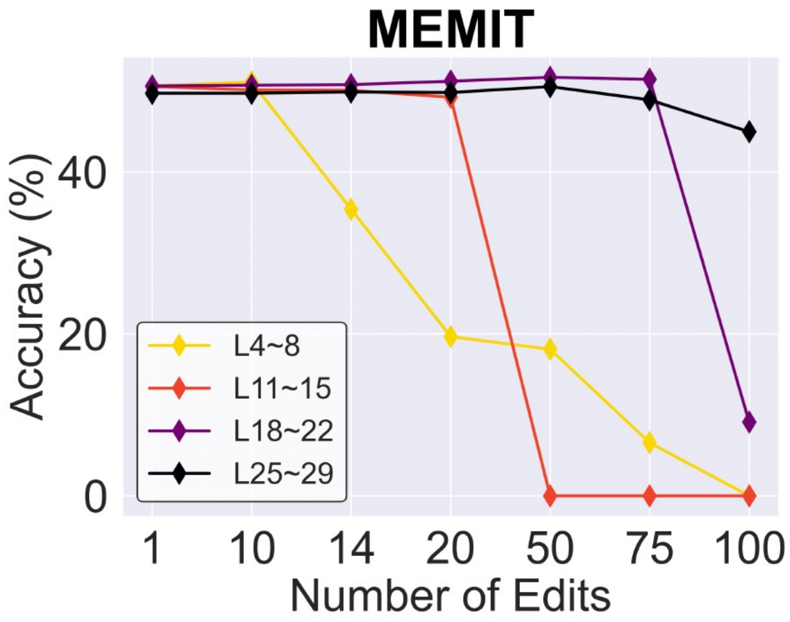
- Methods:
  - ROME: Edit one FFN layer
  - MEMIT: Edit five FFN layers
  - Evaluate the CommonsenseQA dataset.

Layers to Edit

- Methods:
  - ROME: Edit one FFN layer
  - MEMIT: Edit five FFN layers
  - Evaluate the CommonsenseQA dataset.

- The choice of layers for editing in LLMs significantly impacts their general capabilities, <u>with deeper layers showing more resilience to the editing process than shallower ones.</u>





VIRGINIA TECH

Thanks!

Q & A

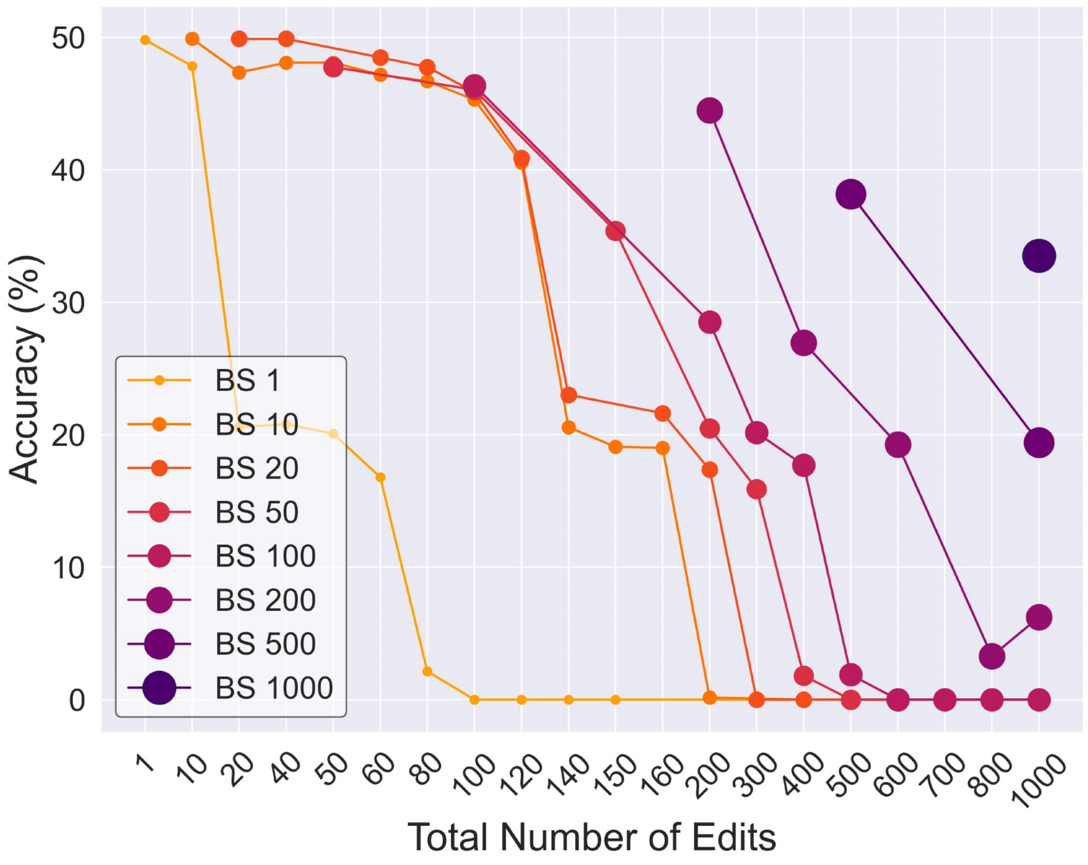
Batch Size of Editing

- Methods:
  - MEMIT: Edit five FFN layers
  - Evaluate the CommonsenseQA dataset.

Batch Size of Editing

- Methods:
  - MEMIT: Edit five FFN layers
  - Evaluate the CommonsenseQA dataset.

- With the same number of edit triples, <u>increasing the batch size means reducing the number of editing times, which turns out to be beneficial in mitigating the damage of ME to LLMs.</u>



VIRGINIA TECH

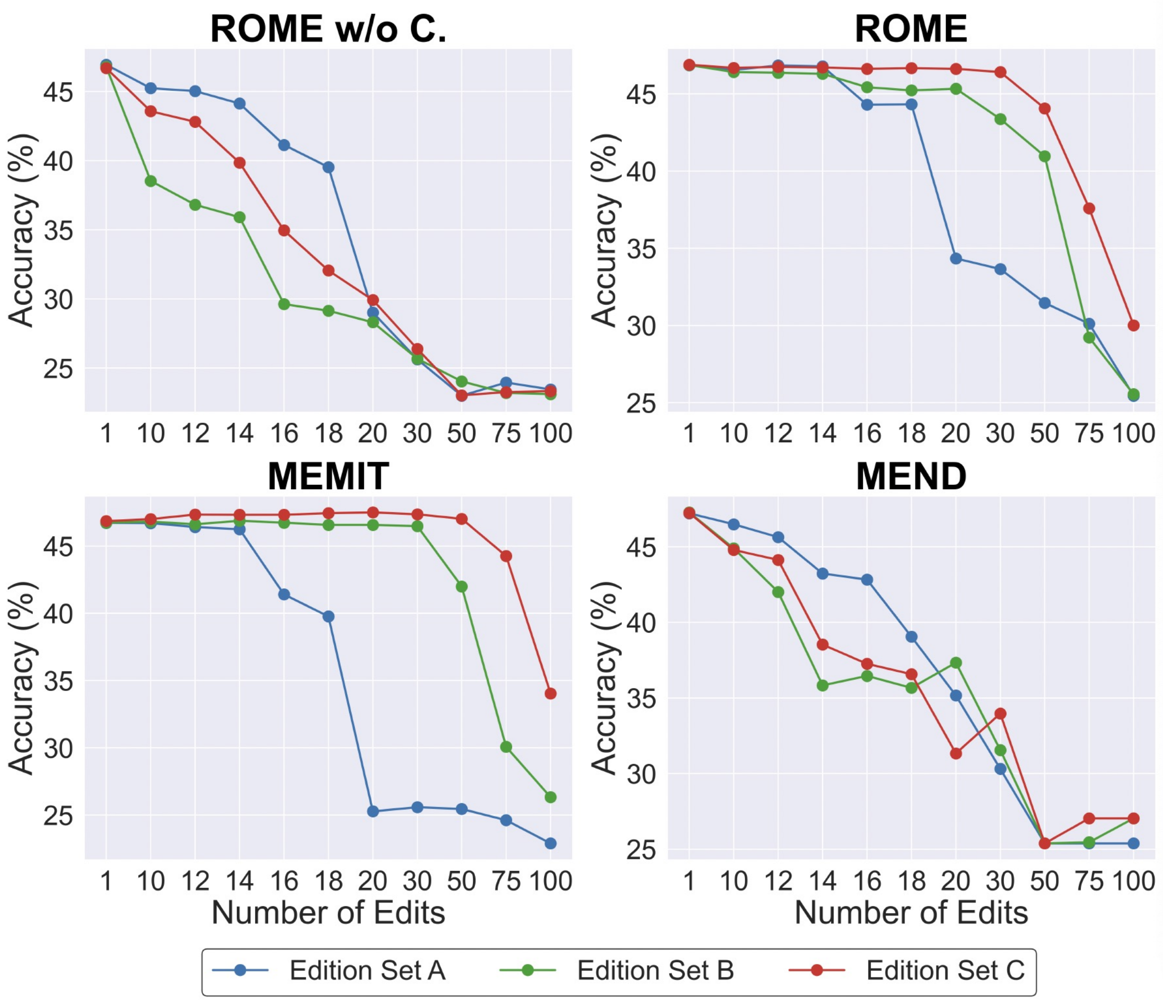
Different Editing Data

- Methods:
  - Randomly select 100 samples from the ZsRE dataset three times without overlapping.
  - Evaluate on CommonsenseQA.

Different Editing Data

- Methods:
  - Randomly select 100 samples from the ZsRE dataset three times without overlapping.
  - Evaluate on CommonsenseQA.

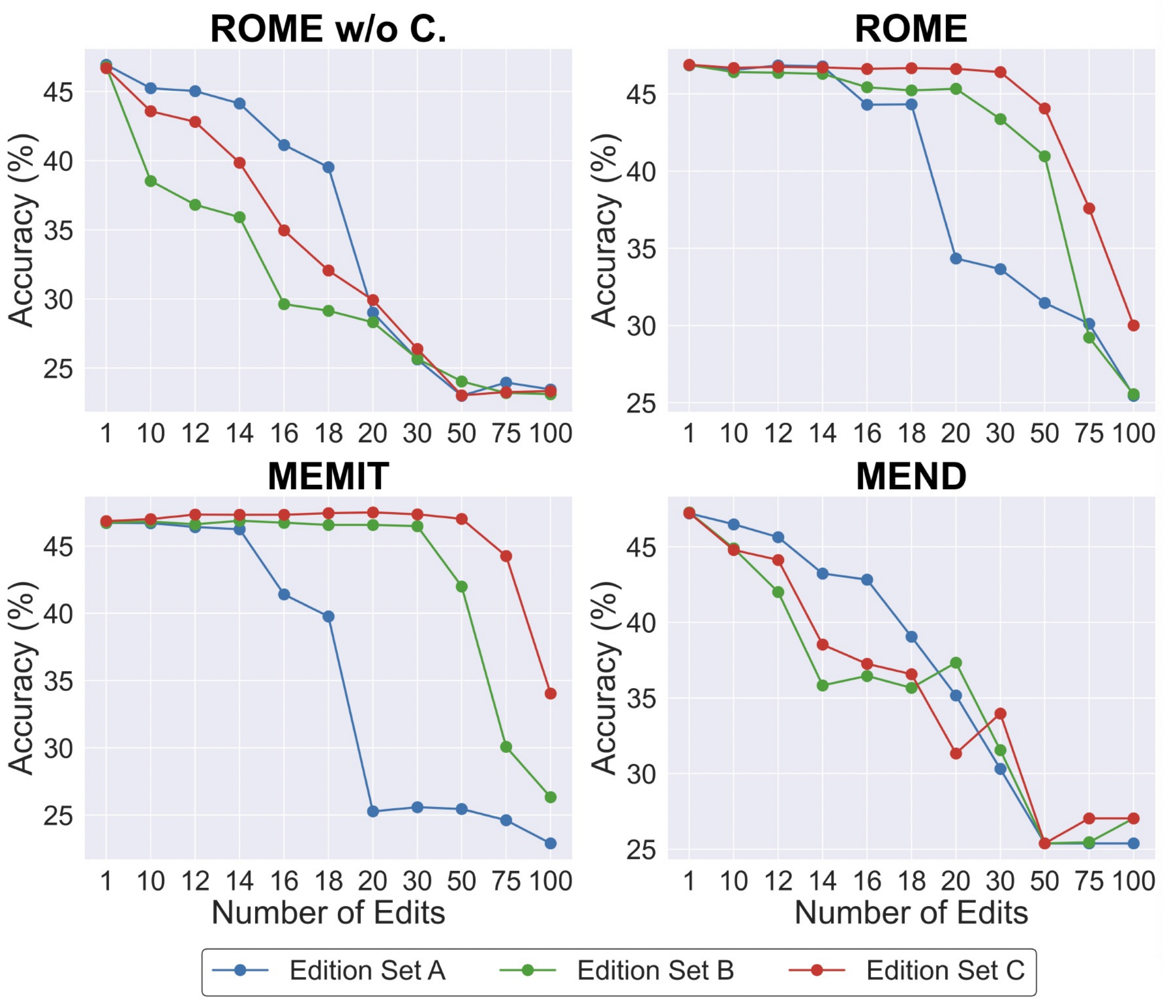
- Under different editing sets, parameter-modifying ME methods systematically destroy the power of the language model after 100 edits.



Different Editing Data

- Methods:
  - Randomly select 100 samples from the ZsRE dataset three times without overlapping.
  - Evaluate on CommonsenseQA.

- <u>The difference in damage trends comes from the effect of editing the data on the model.</u>



References

[1] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300.

[2] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. arXiv preprint arXiv:1811.00937.

[3] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874.

[4] Ahmad Ghazal, Tilmann Rabl, Minqing Hu, Francois Raab, Meikel Poess, Alain Crolotte, and Hans-Arno Jacobsen. 2013. Bigbench: Towards an industry standard benchmark for big data analytics. In Proceedings of the 2013 ACM SIGMOD international conference on Management of data, pages 1197–1208.

[5] OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

[6] Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2020. Investigating prior knowledge for challenging Chinese machine reading comprehension. Transactions of the Association for Computational Linguistics, 8:141–155.

[7] Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically di verse languages. Transactions of the Association for Computational Linguistics, 8:454–470.

[8] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. arXiv preprint arXiv:2108.07732.

[9] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. Fast model editing at scale. In International Conference on Learning Representations.

[10] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in GPT. Advances in Neural Information Processing Systems, 35.

[11] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass editing memory in a transformer. arXiv preprint arXiv:2210.07229.

[12] Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2022. Aging with grace: Lifelong model editing with discrete key-value adaptors. arXiv preprint arXiv:2211.11031.

[13] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timoth.e Lacroix, Baptiste Rozi.re, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.