← Go to **ICML 2024 Conference** homepage (/group?id=ICML.cc/2024/Conference)

MEMORYLLM: Toward Self-Updating Large Language Models

PDF (/pdf?id=p0lKWzdikQ)

Yu Wang (/profile?id=~Yu_Wang24), Yifan Gao (/profile?id=~Yifan_Gao1), Xiusi Chen (/profile?id=~Xiusi_Chen1), Haoming Jiang (/profile?id=~Haoming_Jiang1), Shiyang Li (/profile?id=~Shiyang_Li1), Jingfeng Yang (/profile?id=~Jingfeng_Yang2), Qingyu Yin (/profile?id=~Qingyu_Yin2), Zheng Li (/profile?id=~Zheng_Li9), Xian Li (/profile?id=~Xian_Li3), Bing Yin (/profile?id=~Bing_Yin1), Jingbo Shang (/profile?id=~Jingbo_Shang2), Julian McAuley (/profile?id=~Julian_McAuley1)

Published: 01 May 2024, Last Modified: 01 May 2024 ICML 2024 Conference, Senior Area Chairs, Area Chairs, Reviewers, Publication Chairs, Authors Revisions (/revisions?id=p0lKWzdikQ) BibTeX CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/)

Verify Author List: I have double-checked the author list and understand that additions and removals will not be allowed after the submission deadline.

Keywords: memory, large language models, model editing, long context

TL;DR: MEMORYLLM introduces a novel self-updating architecture in Large Language Models, enabling continuous knowledge integration and retention.

Abstract:
Existing Large Language Models (LLMs) usually remain static after deployment, which might make it hard to inject new knowledge into the model. We aim to build models containing a considerable portion of self-updatable parameters, enabling the model to integrate new knowledge effectively and efficiently. To this end, we introduce MEMORYLLM, a model that comprises a transformer and a fixed-size memory pool within the latent space of the transformer. MEMORYLLM can self-update with text knowledge and memorize the knowledge injected earlier. Our evaluations demonstrate the ability of MEMORYLLM to effectively incorporate new knowledge, as evidenced by its performance on model editing benchmarks. Meanwhile, the model exhibits long-term information retention capacity, which is validated through our custom-designed evaluations and long-context benchmarks. MEMORYLLM also shows operational integrity without any sign of performance degradation even after nearly a million memory updates.

Primary Area: Deep Learning (architectures, generative models, deep reinforcement learning, etc.)

Position Paper Track: No

Paper Checklist Guidelines: I certify that all co-authors of this work have read and commit to adhering to the Paper Checklist Guidelines, Call for Papers and Publication Ethics.

Submission Number: 143

Filter by reply type | Filter by author | Search keywords...

Sort: Newest First | - | = | ≡

Everyone | Program Chairs | Submission143 Authors | Submission143...

21 / 25 replies shown

Submission143 Area... Submission143... Submission143... Submission143...

Submission143... Submission143... Submission143... ✖

Add: Withdrawal

Official Review of Submission143 by Reviewer x2L2

Official Review ✎ Reviewer x2L2 17 Apr 2024, 08:29 (modified: 18 Apr 2024, 12:16)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors, Reviewer x2L2

Revisions (/revisions?id=3qvyZPfPAF)

Summary:
This paper introduces MEMORYLLM, a large language model that incorporates a fixed-size memory pool within its transformer structure to facilitate continuous updates and integration of new information. Unlike traditional static LLMs, MEMORYLLM is designed to manage knowledge dynamically, reducing redundancy and enhancing the model's capacity to adapt over time. The paper provides a detailed discussion of MEMORYLLM's self-update mechanism that selectively refreshes memory tokens, enabling the model to maintain updated and relevant information efficiently.

Extensive evaluations demonstrate MEMORYLLM's performance on model editing and long-context QA benchmarks compared to baseline models. These assessments reveal its ability to handle new information and its robustness in maintaining operational integrity, with no degradation in performance observed after nearly a million updates. The paper also discusses potential applications of MEMORYLLM, suggesting its utility in environments that require continual learning and adaptation. This work opens avenues for further research into efficient memory management and long-term knowledge retention.

Strengths And Weaknesses:
Strengths:

- Impressive results, especially on on quantitative editing and memory efficiency

Weaknesses:

- Reference and relation to prior work is missing, which makes it difficult to judge novelty. This work appears to be a version of Dynamic Evaluation (https://proceedings.mlr.press/v80/krause18a.html) and it is also related to the Fast Weights (https://dl.acm.org/doi/10.5555/3157382.3157582) line of work.
- Clarity: The figures are pretty confusing, and do not aid much in clarifying what is written in the text. Line 42-43 in the first paragraph isn't easy to read either.

Questions:
how do you think performance on long-context scales beyond 16k? 32k and 64k are the now the standard for long context benchmarking?

When selectively updating, are you able to track which tokens in the memory pool contain which pieces of information?

Will the codebase be open-sourced?

Limitations:
The impact section is disappointing and more effort should be dedicated into it.

Ethics Flag: No
Soundness: 3: good
Presentation: 2: fair
Contribution: 3: good
Rating: 5: Borderline accept: Technically solid paper where reasons to accept outweigh reasons to reject, e.g., limited evaluation. Please use sparingly.

Confidence: 5: You are absolutely certain about your assessment. You are very familiar with the related work and checked the math/other details carefully.

Code Of Conduct: Yes

Official Review of Submission143 by Reviewer F5y4

Official Review Reviewer F5y4 20 Mar 2024, 18:31 (modified: 04 Apr 2024, 19:19)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors, Reviewer F5y4

Revisions (/revisions?id=4Y3uoTWprt)

Summary:

The paper presents a novel method to inject new knowledge into large language models using latent representations in the transformer layers as the fixed-size memory pool. The self-update processes integrate the hidden states of the new knowledge and forget old ones with an exponential dropping mechanism. The model was evaluated on benchmarks including model editing tasks and long context question answering tasks, and showed the capacity to retain long-term knowledge and yield robust performance and integrity after extensive memory updates.

Strengths And Weaknesses:

[Strengths]

1. The paper introduces an innovative self-update method to allow LLMs to incorporate and retain new knowledge dynamically even though the weights of LLMs are fixed.
2. The method demonstrates a strong capacity to retain long-term knowledge consistently.
3. MEMORYLLM shows no performance degradation after extensive memory updates. This robustness is important for systems that need frequent updates to reflect the latest information or correct existing data.
4. The authors provided comprehensive evaluations of the model across multiple benchmarks including model editing and long context question-answering tasks.
5. The latent representation of the memory in MEMORYLLM has the potential to be extended to the multimodal domain.

[Weaknesses]

1. Non-redundancy of the knowledge memory is an important aspect that distinguishes MEMORYLLM from prior models that are similar (e.g. [1], [2], [3], [4]). However, there is no quantitative evaluation or experiment on the non-redundancy.
2. The current approach requires memory representation in every transformer layer of the LLM. This design choice would be stronger if the authors could show the necessity and advantage of having memory in every layer through ablation studies.
3. Other baselines are worth considering, such as [5] for the zsRE dataset.
4. The model is based on Llama2 architecture. While I can see the potential to generalize the self-update method to other transformer-based LLMs, it would be ideal to explicitly discuss possible extensions to other language model architectures.
5. What is the scalability of the method in situations where the dataset is large or the system requires rapid/real-time updates of knowledge? What about the generalization of the model across different tasks and domains?

[References]

[1] Zhong W, Guo L, Gao Q, Wang Y. Memorybank: Enhancing large language models with long-term memory. arXiv preprint arXiv:2305.10250. 2023.

[2] Zhang K, Zhao F, Kang Y, Liu X. Memory-augmented llm personalization with short-and long-term memory coordination. arXiv preprint arXiv:2309.11696. 2023.

[3] Zhong Z, Lei T, Chen D. Training language models with memory augmentation. arXiv preprint arXiv:2205.12674. 2022.

[4] Moro G, Ragazzi L, Valgimigli L, Frisoni G, Sartori C, Marfia G. Efficient memory-enhanced transformer for long-document summarization in low-resource regimes. Sensors. 2023.

[5] Glass M, Rossiello G, Chowdhury MF, Gliozzo A. Robust retrieval augmented generation for zero-shot slot filling. arXiv preprint arXiv:2108.13934. 2021.

Questions:

1. Why Figure4 does not have results of several models beyond context length 4k? Specifically, the baseline models with fewer than 7B parameters.
2. What is the possible reason behind the worse performance of MEMORYLLM (as compared to the non-memory baselines) in the qasper dataset?
3. Why use random dropping as the specific forgetting mechanism in the model, not others? For example, applying an exponential decay factor on memory pool from the previous step and aggregating it with the new knowledge.

Limitations:
The authors adequately addressed the limitations and potential negative societal impact of their work.

Ethics Flag: No
Soundness: 3: good
Presentation: 3: good
Contribution: 3: good
Rating: 6: Weak Accept: Technically solid, moderate-to-high impact paper, with no major concerns with respect to evaluation, resources, reproducibility, ethical considerations.
Confidence: 3: You are fairly confident in your assessment. It is possible that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.
Code Of Conduct: Yes

Response to Reviewer F5y4

Official Comment

Authors (Yifan Gao (/profile?id=~Yifan_Gao1), Yu Wang (/profile?id=~Yu_Wang24), Haoming Jiang (/profile?id=~Haoming_Jiang1), Xiusi Chen (/profile?id=~Xiusi_Chen1), +8 more (/group/info?id=ICML.cc/2024/Conference/Submission143/Authors))

27 Mar 2024, 20:49 (modified: 27 Mar 2024, 21:07)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Revisions (/revisions?id=HIGMDVv6RH)

[Deleted]

Rebuttal by Authors

Rebuttal

Authors (Yifan Gao (/profile?id=~Yifan_Gao1), Yu Wang (/profile?id=~Yu_Wang24), Haoming Jiang (/profile?id=~Haoming_Jiang1), Xiusi Chen (/profile?id=~Xiusi_Chen1), +8 more (/group/info?id=ICML.cc/2024/Conference/Submission143/Authors))

29 Mar 2024, 00:23 (modified: 29 Mar 2024, 05:26)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Revisions (/revisions?id=dCy3RAsSYR)

Rebuttal:
Thank you for your detailed comments!
We address the **Weaknesses** part as below:

[W1] Non-redundancy: the non-redundancy mainly comes from the 256 memory tokens in each layer to encapsulate the knowledge in 512 tokens. As shown in Figure 7(c)(d), having 256 memory tokens in each layer performs similarly to 512 memory tokens in each layer, demonstrating the redundancy of raw tokens. We conducted the experiments with 128 memory tokens in each layer, to find that the results were much worse (as mentioned at Line371) thus we chose 256 after these considerations. We report the step 1 accuracy of NQA and SQuAD (see the reference of “step1” in Figure 7) of $K = 128$ here: NaturalQA: 0.34, SQuAD: 0.25. This may serve as the quantitative evaluation of the redundancy and we will add more discussions in the main paper.

[W2] The necessity and advantage of having memory in every layer: We agree that having some ablation studies such as having the memory in only one layer may be helpful. However, it may require a lot of computation to try various memory structures. On top of this paper, we have kept exploring, we tried to augment the model with the memory tokens in a single layer to find that single-layer memory tokens are useless (the validation accuracy on NaturalQA and SQuAD is almost the same with and without the context being injected into the memory). We also tried to augment the model with memory tokens in the last half, the accuracies with one-step update (see the reference of one-step update in Figure 7) are reported below:
MemoryLLM (K=256, current design shown in the paper): NaturalQA: 0.46, SQuAD: 0.39
K=256, only the last half layers are augmented with memory tokens: NaturalQA: 0.39, SQuAD: 0.22.
This shows that having the memory tokens in both the first half and the second half layers are necessary for better performance.

[W3] Other baselines: We want to point out that the model editing benchmarks are using ZsRE dataset but this dataset has been adjusted for model editing purposes, thus the task has also been adjusted, and methods such as [5] may not be easily adjusted for model editing tasks.

Replying to Rebuttal by Authors

Rebuttal by Authors

Rebuttal

Authors (Yifan Gao (/profile?id=~Yifan_Gao1), Yu Wang (/profile?id=~Yu_Wang24), Haoming Jiang (/profile?id=~Haoming_Jiang1), Xiusi Chen (/profile?id=~Xiusi_Chen1), +8 more (/group/info?id=ICML.cc/2024/Conference/Submission143/Authors))

29 Mar 2024, 00:24 (modified: 29 Mar 2024, 05:26)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Revisions (/revisions?id=4asxBzj6th)

Rebuttal:
[W4] Extension to other architectures: Our framework is indeed generalizable to any LLMs with transformer architectures and full attention mechanism in the transformer. In this paper, we mainly aim to propose a model that is self-updatable and has long-term knowledge retention thus we chose Llama2 as this is the most popular model. We would add the discussion into our paper clarifying that we are proposing MemoryLLM while the framework is also generalizable to other LLMs.

[W5] Scalability/Rapid or Real-time change: One advantage of our model is that the memory size can be scalable to encapsulate more knowledge. Meanwhile, during injection, we do not need to use the whole memory, thus the injection efficiency is not affected by the larger size of the memory pool. The generalizability of the MemoryLLM is mainly determined by the generalizability of the backbone large language model and the dataset. We believe that with enough datasets and enough training, generalizability across different domains can be expected.

We answer the questions as follows:

[Q1] Results of models in the situations with context length longer than 4k: Llama2 has the maximum context window as 4k, where the performance would drop to zero once it surpasses the context length.

[Q2] Performance degradation on Qasper: We perform our training of MemoryLLM on C4 dataset, whereas the dataset RedPajama used for training Llama2 has a big proportion of Arxiv dataset. Thus, the training may affect the model's ability on scientific datasets (such as Qasper). We did not expect this scenario when training the model, so in the next generation of MemoryLLM (which we are working on now), we will involve RedPajama-V2 as the training set and try to make the training dataset more balanced.

Replying to Rebuttal by Authors

Rebuttal by Authors

Rebuttal

Authors (Yifan Gao (/profile?id=~Yifan_Gao1), Yu Wang (/profile?id=~Yu_Wang24), Haoming Jiang (/profile?id=~Haoming_Jiang1), Xiusi Chen (/profile?id=~Xiusi_Chen1), +8 more (/group/info?id=ICML.cc/2024/Conference/Submission143/Authors))

29 Mar 2024, 00:25 (modified: 29 Mar 2024, 05:26)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Revisions (/revisions?id=1gEA2csGwz)

Rebuttal:

[Q3] Random Dropping: random dropping is a fairly straightforward way to keep the size of the memory pool fixed while maintaining an exponential forgetting mechanism. Applying the exponential decay factor on the memory pool from the previous step is a possible way but the aggregating might be simply adding the old memory with the new memory. During our explorations we also tried aggregation instead of random dropping, however, we found that keeping the integrity of hidden states of the tokens seems to be beneficial while adding hidden states may usually affect both the original knowledge and the new knowledge, where we concluded that adding hidden states may not be a natural way to integrate different hidden states.

[5] Glass M, Rossiello G, Chowdhury MF, Gliozzo A. Robust retrieval augmented generation for zero-shot slot filling. arXiv preprint arXiv:2108.13934. 2021.

Replying to Rebuttal by Authors

Official Comment by Reviewer F5y4

Official Comment Reviewer F5y4 04 Apr 2024, 10:34

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Comment:

Thank you for the detailed responses. It would be very helpful to include some of these discussions in the updated paper. Overall, I would keep my positive rating.

Response to Reviewer F5y4

Official Comment

Authors (Yifan Gao (/profile?id=~Yifan_Gao1), Yu Wang (/profile?id=~Yu_Wang24), Haoming Jiang (/profile?id=~Haoming_Jiang1), Xiusi Chen (/profile?id=~Xiusi_Chen1), +8 more (/group/info?id=ICML.cc/2024/Conference/Submission143/Authors))

04 Apr 2024, 18:07 (modified: 04 Apr 2024, 18:07)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Revisions (/revisions?id=UO4o5bH58r)

Comment:

Dear Reviewer F5y4,

Thank you for your positive and constructive feedback. In response to your suggestions, we've thoroughly revised our paper and added these discussions. Could we kindly ask you to consider re-evaluating the score in light of these improvements? We value your expertise and are eager to hear any additional feedback you may have.

Warm regards,
Authors

Replying to Response to Reviewer F5y4

Official Comment by Reviewer F5y4

Official Comment Reviewer F5y4 04 Apr 2024, 19:20

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Comment:
Yes! You have addressed my major concerns in the rebuttal.

Some additional suggestions: It would be especially nice to include the redundancy evaluations with varying K under larger N settings; the accuracies reported in response to [W2]; and, related to [Q1], explanations of MemoryLLM vs LongLlama/OpenLlama performance in 2wikimqa dataset within 4k (or even 2k) window.

I have increased my rating accordingly. Looking forward to seeing the final paper.

Official Review of Submission143 by Reviewer d7sJ

Official Review Reviewer d7sJ 20 Mar 2024, 14:36 (modified: 21 Mar 2024, 05:11)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors, Reviewer d7sJ

Revisions (/revisions?id=sjEFV9x06K)

Summary:
Authors tackle the problem of how to reduce the cost of keeping a model updated once its has been trained. The existing mechanisms of updating data include RAG (augmenting information at runtime, depends on the effectiveness of techniques such as vector search), editing models, and using longer contexts.

Authors introduce a mechanism to allow the model to self-update with new knowledge, while minimizing degradation of previously learned knowledge.

Strengths And Weaknesses:
Strengths:

1. The problem statement is easy to understand and impactful if solved.
2. The solution proposed is novel and non convoluted.

Weaknesses

1. The paper doesn't talk about the safeguards and checks we should put in place to ensure that model quality doesnt drift over learning cycles (and how to get the quality back on track if it drifts).

Questions:

1. There is a risk that a model performance significantly drifts after a "knowledge update" loop, either in terms of quality of answers, or in terms of safety, latency etc. How do we safeguard against such drifts at training time?
2. Can the process of self-update be fully automated? For example, can the model automatically infer when it should update itself with new knowledge?

Limitations:
The main limitation is the risk of the model drift -- as it updates, the model might lose quality, latency or fairness. As such, deploying this updated model will require non trivial checks and balances, which will slow down the overall process of self-update and deployment into production.

Ethics Flag: No
Soundness: 3: good
Presentation: 2: fair
Contribution: 3: good
Rating: 6: Weak Accept: Technically solid, moderate-to-high impact paper, with no major concerns with respect to evaluation, resources, reproducibility, ethical considerations.

Confidence: 4: You are confident in your assessment, but not absolutely certain. It is unlikely, but not impossible, that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work.

Code Of Conduct: Yes

Response to Reviewer d7sJ

Official Comment

Authors (Yifan Gao (/profile?id=~Yifan_Gao1), Yu Wang (/profile?id=~Yu_Wang24), Haoming Jiang (/profile?id=~Haoming_Jiang1), Xiusi Chen (/profile?id=~Xiusi_Chen1), +8 more (/group/info?id=ICML.cc/2024/Conference/Submission143/Authors))

27 Mar 2024, 20:51 (modified: 27 Mar 2024, 21:08)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Revisions (/revisions?id=IzQaO9DwkB)

[Deleted]

Rebuttal by Authors

Rebuttal

Authors (Yifan Gao (/profile?id=~Yifan_Gao1), Yu Wang (/profile?id=~Yu_Wang24), Haoming Jiang (/profile?id=~Haoming_Jiang1), Xiusi Chen (/profile?id=~Xiusi_Chen1), +8 more (/group/info?id=ICML.cc/2024/Conference/Submission143/Authors))

29 Mar 2024, 00:25 (modified: 29 Mar 2024, 05:26)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Revisions (/revisions?id=gBYg7gG6NQ)

Rebuttal:

Thank you so much for your appreciation of our work! We address your concerns below:

Model performance drifting over learning cycles: We agree this is a very important problem and we spent an enormous time on this problem when designing the structure. We aim to mention two points here: (1) Our memory pool is constantly being updated during training, so the model needs to guarantee that the representation of every layer should be of the same distribution, as this is the only way for the model to be able to recognize the nearest memory tokens from the next layer (note that the outputted memory tokens in layer $l+1$ would be used as the memory tokens in layer l). With billions or even trillions of updates during training, the model should strictly follow this paradigm, which offers the potential of many steps of updates without drifting. (2) The transformer weight is fixed during inference, thus with the distribution of the memory tokens being consistent, we think it is possible to maintain the model's quality, latency, and fairness over many updates.

Can the process of self-update be fully automated? Currently, we need to give the context to the model and ask the model to update according to the context. We agree that having the model itself determine if it needs to absorb the context could be meaningful, however, with the intuition of human memory, when we are encountering and seeing things, these things are injected into our memory without our own decisions. Our eventual goal is to have a self-updatable model working like human memory, where whether automated self-update is needed may be debatable. We will keep exploring and thinking about this question when our model becomes more and more powerful in the future.

Official Review of Submission143 by Reviewer XDyx

Official Review Reviewer XDyx 17 Mar 2024, 16:44 (modified: 21 Mar 2024, 05:11)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors, Reviewer XDyx

Revisions (/revisions?id=Ck2NqSYPJQ)

Summary:
The paper proposes MEMORYLLM, a novel self-updatable large language model that incorporates a fixed-size memory pool within its transformer layers. The key idea is to divide the model parameters into static and updatable parts, allowing efficient knowledge integration without full retraining. The authors augment a 7B parameter Llama2 model with a 1B parameter memory pool and devise techniques for updating and training this memory. Extensive experiments demonstrate MEMORYLLM's ability to effectively absorb new knowledge and maintain robustness.

Strengths And Weaknesses:
Strengths:

- Scalable approach with fixed-size memory, avoiding unbounded growth issues of retrieval-based or context-based methods.
- Strong performance across diverse tasks like model editing, long context QA, and knowledge retention.

Weaknesses:

- Limited exploration of extremely large memory pool sizes due to computational constraints.
- No comparisons to very recent methods like PrefixTuning or instruction-tuned models on knowledge editing tasks.

Questions:

- How would the memory update mechanism need to be modified to handle contradictory or conflicting knowledge inputs? Does the current scheme have provisions for consistency checking?
- While the discrete update steps allow efficient knowledge editing, is there a way to enable more continuous or online learning within the MEMORYLLM framework? This could be useful for domains with rapidly evolving information streams.

Limitations:
No potential negative societal impact found.

Ethics Flag: No
Soundness: 3: good
Presentation: 4: excellent
Contribution: 3: good
Rating: 6: Weak Accept: Technically solid, moderate-to-high impact paper, with no major concerns with respect to evaluation, resources, reproducibility, ethical considerations.
Confidence: 3: You are fairly confident in your assessment. It is possible that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work. Math/other details were not carefully checked.
Code Of Conduct: Yes

Response to Reviewer XDyx

Official Comment

Authors (Yifan Gao (/profile?id=~Yifan_Gao1), Yu Wang (/profile?id=~Yu_Wang24), Haoming Jiang (/profile?id=~Haoming_Jiang1), Xiusi Chen (/profile?id=~Xiusi_Chen1), +8 more (/group/info?id=ICML.cc/2024/Conference/Submission143/Authors))

27 Mar 2024, 20:55 (modified: 27 Mar 2024, 21:08)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Revisions (/revisions?id=C9T9WI8wpd)

[Deleted]

Rebuttal by Authors

Rebuttal

Authors (Yifan Gao (/profile?id=~Yifan_Gao1), Yu Wang (/profile?id=~Yu_Wang24), Haoming Jiang (/profile?id=~Haoming_Jiang1), Xiusi Chen (/profile?id=~Xiusi_Chen1), +8 more (/group/info?id=ICML.cc/2024/Conference/Submission143/Authors))

29 Mar 2024, 00:25 (modified: 29 Mar 2024, 05:26)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Revisions (/revisions?id=bhb8N0pKJi)

Rebuttal:
Thank you so much for your comments and suggestions! We address your concerns on **Weaknesses** as below:

[W1] Limited exploration of extremely large memory pool due to computation constraints: We aim to explore possible efficient training strategies to enable a larger memory pool. As this is not trivial and may require more exploration, we put it in our future endeavors.

[W2] PrefixTuning or Instruction-tuned models on model editing benchmarks: We found a most recent work about model editing [1] that is using purely finetuning which we think may be related to what you describe. This is highly related to our task, however, it was released on Arxiv on March 10th, 2024, which may serve as concurrent work as ours. We will compare with this method when proposing the next generation of MemoryLLM.

We answer the questions below:

[Q1] Handling contradictory knowledge inputs: Currently, we model the input stream as the form of continuous context, and we shard long context into several parts and input them into the model for the loss calculation on the last part, which is similar to long context finetuning. As the conflicting knowledge is not deemed a problem for long context models, we hope our model could have the same property when having long context inputs.

[Q2] Continuous or Online Learning: The eventual goal of this project is to have a model that could interact with humans while also learning from the interactions. The learning may result in the update of the memory parameters and the parameters of the transformer. To enable this property we believe that expanding the memory size and longer knowledge retention would be crucial, this is the direction we are currently exploring.

Official Review of Submission143 by Reviewer A72y

Official Review Reviewer A72y 14 Mar 2024, 07:07 (modified: 02 Apr 2024, 06:10)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors, Reviewer A72y

Revisions (/revisions?id=oDfAb7pcwQ)

Summary:
This paper presents MemoryLLM, a technique that efficiently updates the knowledge of the language model based on the given input. Specifically, MemoryLLM compresses the input into a memory token (similar to a KV cache), where this memory token is updated using the text knowledge. Authors considered multiple applications to show the MemoryLLM retains the knowledge of past while effectively learning new knowledge.

Strengths And Weaknesses:
Strengths

The author considered multiple experimental setups.

The overall writing is clear and well-organized.

The method is sound, and the ablation study well supports the idea's strength.

Weakness

[W1] Limited experimental support of the major claim. The major claim of the paper is that the proposed method compensates for the limitation of retrieval augmentation, model editing, and long context methods. However, the overall experiments do not support this claim. Furthermore, the related works are not discussed well.\

[W1-1] Comparison with retrieval augmentation methods.
First, experimental comparison with retrieval augmentation methods [1,2] is missing. Also, the claim in the (first) introduction paragraph should be modified as LLM does not require saving the image. For me, it is hard to believe that saving raw text requires lots of storage compared to the current method (as it stores the latent). I would like to ask the authors for the direct storage comparison of raw text and the proposed method.

[W1-2] Comparison with more recent and relevant works is required. For instance, SERAC [3] and GRACE [4] utilize memory banks for model editing. Since there are similar lines of research, I politely request the authors to discuss the papers in-depth and also compare them during the rebuttal

[W1-3] Furthermore, there are related works that use memory banks to tackle long-context understanding [5].

[W2] Also, in Figure 4, LLama often shows better results than the proposed method. Since the proposed method uses more computation and parameters, I believe the results should (at least) outperform LLama. I do agree that LongBench is aiming out-of-context window, but the performance within the context window is also important.

[W3] While the paper claims the proposed method is efficient, there is no efficiency comparison with others.

[Q1] It would be interesting to see the effect of random dropping. i) not dropping the old memories (expanding memories), ii) better strategy than random dropping, since there will be memories that are less important (or less frequently accessed) than others.

[Q2] Are there any applications that are not using QA? The reason is that retrieval augmentation is a nice option for QA tasks when continual learning (or knowledge updating). If there are other tasks (than QA) where retrieval augmentation is not effective and memoryllm is effective, it will be very interesting.

Overall, I think the author tried multiple applications (which is good), but needs to focus more on one special case (e.g., model editing, long context). Currently, it is hard to believe that it outperformed the baselines in all domains, as there are several missing baselines.

The references are below
[1] BM25
[2] Dense Passage Retrieval for Open-Domain Question Answering, EMNLP 2020
[3] Memory-based model editing at scale, ICML 2022
[4] Aging with grace: Lifelong model editing with discrete key-value adaptors, NeurIPS 2023
[5] Augmenting language models with long-term memory, NeurIPS 2023

Questions:
See the question above

Limitations:
The authors discussed the future work.

Ethics Flag: No
Soundness: 2: fair
Presentation: 3: good
Contribution: 2: fair
Rating: 5: Borderline accept: Technically solid paper where reasons to accept outweigh reasons to reject, e.g., limited evaluation. Please use sparingly.
Confidence: 4: You are confident in your assessment, but not absolutely certain. It is unlikely, but not impossible, that you did not understand some parts of the submission or that you are unfamiliar with some pieces of related work.
Code Of Conduct: Yes

- = ≡

Rebuttal by Authors

Rebuttal

✎ Authors (👁 Yifan Gao (/profile?id=~Yifan_Gao1), Yu Wang (/profile?id=~Yu_Wang24), Haoming Jiang (/profile?id=~Haoming_Jiang1), Xiusi Chen (/profile?id=~Xiusi_Chen1), +8 more (/group/info?id=ICML.cc/2024/Conference/Submission143/Authors))

📅 29 Mar 2024, 00:51 (modified: 29 Mar 2024, 05:26)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Revisions (/revisions?id=j5uS3ZAIhL)

Rebuttal:

Thanks for the detailed comments, we address the concerns below:

[W1-1] Comparison with RAG methods: The primary goal of MemoryLLM is to achieve self-updatable LLM where the memory module serves as the parameters that could keep updating along the inference process, whereas RAG methods aim to retrieve the most relevant piece of information from the history. Intuitively, RAG is used to conduct coarse retrieval from millions of documents, while the retrieved documents can be processed by MemoryLLM. We use BM25 retriever to extract 4k tokens from the whole context and use MemoryLLM to process these 4k tokens to generate the answer. The results are shown below:

	MemoryLLM-7b-16k	MemoryLLM-7b-all-BM25
narrativeqa	**20.64**	15.6
qasper	19.57	**20.3**
multifieldqa_en	29.56	**33.08**
hotpotqa	**34.03**	32.27
2wikimqa	**27.22**	24.17
musique	13.47	**15.36**

Here `MemoryLLM-7b-16k` corresponds to the results in Figure 4, and `MemoryLLM-7b-all-BM25` means retrieving 4k tokens from the whole given context and using MemoryLLM to process the retrieved 4k tokens. From the results, we can see that using the BM25 retriever could enhance the model performance on certain datasets while not universally beneficial.

[W1-2] Comparison with mass editing methods: The reason that we did not compare with SERAC and GRACE is that they both target at mass editing, where SERAC can make around 75 edits without performance dropping, and GRACE can make around 1000 edits. We also noticed that MEMIT[6] could make 10000 edits. These methods are impressive, however, when doing single-fact editing, they do not show better performance than the methods that are specifically designed to edit one single fact. Model editing could serve as a downstream task of MemoryLLM, however, MemoryLLM is not specifically designed for model editing tasks.

Rebuttal by Authors

Rebuttal

Authors (Yifan Gao (/profile?id=~Yifan_Gao1), Yu Wang (/profile?id=~Yu_Wang24), Haoming Jiang (/profile?id=~Haoming_Jiang1), Xiusi Chen (/profile?id=~Xiusi_Chen1), +8 more (/group/info?id=ICML.cc/2024/Conference/Submission143/Authors))

29 Mar 2024, 00:51 (modified: 29 Mar 2024, 05:26)

Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Revisions (/revisions?id=E3kFugJyHe)

Rebuttal:

[W1-3] Comparison with LONGMEM: LONGMEM[5] is indeed related and we include that in our related work. The reason that we did not compare with LONGMEM is that their major results are on perplexities, few-shot learning, and chapterbreak, and their method is implemented based on GPT2. As the behavior of GPT2 is very different from Llama2, it may be hard to expect the performance of LONGMEM to be similar to Llama2 as on GPT2. We are committed to comparing with them if they have published the code on Llama2.

[W2] Comparison with Llama2 within the 4k context window: We agree that the performance of our model should align with Llama2 when the context length is smaller than 4096. We would like to mention that with context length=4k, our model is better than Llama2 on 2wikimqa, musique, and comparable with Llama2 on narrativeqa, worse than Llama2 on qasper, multifieldqa, and hotpotqa. Compared with the Red-Pajama dataset used for training Llama, we did not include the Arxiv dataset, which may affect the performance of our model on the benchmarks related to scientific articles (especially Qasper), while the ability of our model may be enhanced on some other datasets.

[W3] Efficiency Comparison: Our efficiency mainly lies in the fact that the update process only takes forward without any backpropagation. During self-update, our model may be more efficient than using finetuning or continually training to update the model. For RAG and long context methods, it is hard to compare the efficiency as they do not need any update, while our model keeps updating when injecting the knowledge. Besides, as previous methods mostly do not support self-updating, this process is hard to evaluate compared with other models. As there are more parameters during inference, the generation process is not more efficient than Llama2 itself. We will add more clarifications about efficiency in the paper.

[5] Augmenting language models with long-term memory, NeurIPS 2023
[6] Mass-Editing Memory in a Transformer, ICLR 2023

Replying to Rebuttal by Authors

Rebuttal by Authors

Rebuttal
Authors (Yifan Gao (/profile?id=~Yifan_Gao1), Yu Wang (/profile?id=~Yu_Wang24), Haoming Jiang (/profile?id=~Haoming_Jiang1), Xiusi Chen (/profile?id=~Xiusi_Chen1), +8 more (/group/info?id=ICML.cc/2024/Conference/Submission143/Authors))
29 Mar 2024, 00:58 (modified: 29 Mar 2024, 05:26)
Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors
Revisions (/revisions?id=yiHqT7xonX)

Rebuttal:
We address the questions below:

[Q1] Effects of Random Dropping: Since the size of our memory pool is fixed during training, it may be hard to keep all the memory tokens without dropping as it will change the size of the memory. We could achieve stable updating (the model is functioning properly even after several hundreds of thousands of updates) mainly due to the enormous training with different memory tokens but of the same size. Expanding memories may lose this property and may require redesigning of the model architecture. We agree that “there will be memories that are less important (or less frequently accessed) than others.” so we are also trying to propose better policies to drop memory tokens in the successive exploring of this paper.

[Q2] Possible evaluations other than QA: If there are benchmarks that have long context input while the question is related to the whole context rather than a small piece of the context, RAG may not perform well while long context methods and MemoryLLM could perform better. To this end, we run some experiments on the Qmsum benchmark:

	Llama2-7b-4k	Llama2-7b-4k-BM25	MemoryLLM-7b-16k
qasper	21.29	20.64	20.67

Here `Llama-7b-4k` means truncating the input context into 4k tokens, `Llama2-7b-4k-BM25` refers extracting 4k tokens from the whole context using BM25 retriever. `MemoryLLM-7b-16k` is our model with an input context length of 16k. Here we observe that MemoryLLM achieves similar performances as RAG. This means on tasks where the whole context needs to be used, RAG may not achieve better performances.

Replying to Rebuttal by Authors

Thank you for the rebuttal

Official Comment Reviewer A72y 02 Apr 2024, 06:11
Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Comment:
I thank the authors for addressing my concerns throughout the rebuttal. I have updated my score accordingly.

Replying to Thank you for the rebuttal

Thank you for your constructive comments and response

Official Comment

Authors (Yifan Gao (/profile?id=~Yifan_Gao1), Yu Wang (/profile?id=~Yu_Wang24), Haoming Jiang (/profile?id=~Haoming_Jiang1), Xiusi Chen (/profile?id=~Xiusi_Chen1), +8 more (/group/info?id=ICML.cc/2024/Conference/Submission143/Authors))

02 Apr 2024, 21:03 Program Chairs, Senior Area Chairs, Area Chairs, Reviewers Submitted, Authors

Comment:
We sincerely thank the reviewer for the constructive comments and instant response. Thank you so much!