Mastering Long-Context Multi-Task Reasoning With Transformers and Recurrent Memory

A. Bulatov,[1,2,*] Y. Kuratov,[1,2,†] and M. Burtsev[3,‡]

[1] *AIRI*

[2] *Neural Networks and Deep Learning Lab, MIPT*

[3] *London Institute for Mathematical Sciences*

Abstract. Recent advancements have significantly improved the skills and performance of language models, but have also increased computational demands due to the increasing number of parameters and the quadratic complexity of the attention mechanism. As context sizes expand into millions of tokens, making long-context processing more accessible and efficient becomes a critical challenge. Furthermore, modern benchmarks such as BABILong [1] underscore the inefficiency of even the most powerful LLMs in long context reasoning. In this paper, we employ finetuning and multi-task learning to train a model capable of mastering multiple BABILong long-context reasoning skills. We demonstrate that even models with fewer than 140 million parameters can outperform much larger counterparts by learning multiple essential tasks simultaneously. By conditioning Recurrent Memory Transformer [2] on task description, we achieve state-of-the-art results on multi-task BABILong QA1-QA5 set for up to 32k tokens. The proposed model also shows generalization abilities to new lengths and tasks, along with increased robustness to input perturbations.

Keywords: *deep learning, recurrent neural networks, memory-augmented neural networks, long context processing*

*Electronic address: bulatov.as@phystech.edu

†Electronic address: yurii.kuratov@phystech.edu,

‡Electronic address: mb@lims.ac.uk

I. INTRODUCTION

The field of language modeling has significantly advanced in recent years. Introduction of the Transformer [3] architecture and openly available large language models (LLMs) such as GPT-4 [4], Claude [5] and Gemini [6] made intelligent assistants much more smart, helpful and able to process increasingly hard tasks. The abilities of language comprehension improve with the number of parameters for these models, which makes processing information increasingly time-consuming and expensive. Additionally, the quadratic complexity of attention amplifies the problem when the context size of the task increases.

The problem of long-context processing becomes even more relevant with input sizes of models growing up to 32k or 128k tokens, and some models even reaching the verge of millions [6] and tens of millions tokens [1]. Most of these methods still require computational resources that are unavailable to most users. Thus, making long-context processing more accessible and efficient is a critical challenge.

Approaches combining language models and recurrent neural networks have achieved remarkable success in solving individual long-context tasks. The recent BABILong benchmark [1] shows that with proper finetuning, Recurrent Memory Transformer (RMT) [2] and Mamba [7] models with less than 140M parameters can outperform counterparts with tens of billions of parameters. However, in previous works, these models were trained only on individual BABILong tasks, and the possibility of mastering the whole set of skills at once remains unexplored.

BABILong is built on top of the bAbI [8] benchmark. It was proposed in 2016 with the aim of developing a set of proxy tasks, a prerequisite for any system aiming at full-text understanding. BABILong can be seen as its long-context analog, which represents minimal requirements for an LM-based assistant with extensive input size. Despite considerable success on bAbI, even the strongest contemporary LLMs fail to retain the high performance on BABILong when the context length is increased.

In this work, we explore the possibility of training an agent to master multiple BABILong reasoning skills simultaneously, rather than training separate models for each skill. We demonstrate that this is possible with Recurrent Memory Transformer, based on GPT-2 [9] with only 137 million parameters. Additionally, we investigate the effects of multi-task training along with the mutual effects of BABILong skills on each other to prove their

diversity and comprehensiveness.

A. Contributions

Our main contributions are the following:

1. We propose an efficient input arrangement for RMT adding the question at the beginning of a text, that enhances multi-task accuracy up to 12% on average as well as single-task performance.

2. By exploiting multi-task learning with RMT, we achieve the best performance on QA1-QA5 BABILong tasks among multi-task models, which holds up for task lengths up to 32k tokens, outperforming GPT-4 and other large language models.

3. We analyze the influence of individual tasks on multi-task learning, demonstrating the improved ability of multi-task RMT to generalize to new lengths and tasks.

4. We evaluate the robustness of RMT to changes in names and objects in BABILong, showing that the multi-task version remains more robust with input perturbations.

II. RELATED WORK

With the emergence and growing success of Transformer models [3], improving their efficiency for long-context processing became a highly popular research area. The earlier works such as Longformer [10], Big Bird [11], and more recent LongNet [12] use the sparse attention approach that aims at reducing the number of computed attention values. Another research direction studies the ways of improving the architecture and training methods for recurrent neural networks to reduce the complexity from quadratic to linear. RWKV [13], Hawk, Griffin [14] and xLSTM [15] combine recurrence with efficient parallelization, other models use the state space modeling approach: S4 [16], Mamba [7] and others.

A compromise between attention and token-wise recurrence is segment-wise recurrence. Transformer-XL [17] splits the sequence into manageable segments and processes them one by one. The memory in this case is represented by the full cache of hidden states. Memformer [18] introduces additional memory storage and a module that performs memory

operations. Recurrent Memory Transformer [2] provides a memory storage comprised of tokens, and reading and writing from the memory is performed by the model itself.

Augmenting neural networks with memory is often used to increase their efficiency and performance along with recurrence. From the early works [19, 20], neural networks were combined with memory, later enhanced with the introduction of the Backpropagation Through Time learning algorithm [21] and the Long-Short Term Memory (LSTM) [22] architecture. More recent works such as Neural Turing Machines [23] and Memory Networks [24] were considered as alternatives to general recurrent models, evaluated on the bAbI benchmark [8].

Multiple works introduced benchmarks [25] and investigated the comprehensive set of language model abilities across various scenarios [26–28]. As the context length of language models increased, so did the benchmark lengths. In recent years a number of datasets aiming at evaluating long context were released: LongAlign and LongBench-chat [29], ZeroScrolls, LongICLBench [30], L-Eval [31]. However, natural text benchmarks are generally too short and do not suffice for evaluating the enormous contexts of Gemini and RMT. This motivated the creation of generative benchmarks with scalable lengths of tasks: simple but insightful "needle-in-a-haystack" LLMTest [1] with magic numbers as needles in Paul Graham essays as a haystack; passkey and key-value retrieval tasks are part of InfinityBench [32], and RULER [33] introduces multiple types of "needles". The BABILong [1] benchmark goes further to put a variety of reasoning skills to test in the long-context scenario.

III. BABILONG LONG-CONTEXT BENCHMARK

BABILong [1] is a long-context benchmark, with 20 diverse tasks that tackle various aspects of reasoning, including fact chaining, simple induction, deduction, counting, handling lists, sets, and others. The BABILong samples are created by embedding the generative bAbI [8] tasks within irrelevant sentences of background text, sampled from the PG-19 [34] corpus. By varying the amount of background text, each task can be scaled to different lengths from 0k, denoting raw bAbI tasks with no background text, up to 10 million tokens, and beyond.

The BABILong tasks vary in difficulty and the required aspect of reasoning. The QA1

[1] `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`

"single supporting fact" task requires answering a question about a person's location using a single supporting fact. The QA2 "two supporting facts" and QA3 "three supporting facts" introduce the challenge of differentiating subjects and objects, utilizing two and three supporting facts, respectively. The QA4 "two-argument relation" tackles spatial reasoning through two-argument relations, while the QA5 "three-argument relation" task involves tracking multiple objects to solve the three-argument relation problem. A more detailed description of each task is provided in the bAbI paper [8].

For finetuning we format the model input using the following pattern and experiment with repeating the question at the beginning and the end of the input.

1. `{Context}\n\nQuestion: {Question}`
2. `Question: {Question} {Context}\n\nQuestion: {Question}`

In our previous work, we used the first option with the question at the end. This is a very demanding setup because a model has no information about the specifics of the question when processing the context, as it will only be available at the end. Placing the question at the beginning allows the model to be conditioned on the task and potentially should result in more efficient processing of the context. In this work, we introduce this option and adopt it in experiments by default. For LLM evaluation the first pattern is used, and the instruction, in-context examples, and post prompt are added in the beginning.

IV. MULTI-TASK TRAINING AND EVALUATION

We train RMT on the first ten tasks from QA1 to QA10, leaving the other tasks starting from QA11 as a test for out-of-domain performance. For multi-task training, we choose the task randomly for every batch. For RMT, the segment size is set to 512 tokens, and memory consists of 16 tokens.

Following the original work [1], we use curriculum training with a sequentially increasing number of segments. RMT uses the following schedule for the number of segments: 1-2-4-6-8-16-32, meaning the training is stopped at 32 segments or 16k tokens. During each curriculum stage n the number of segments is chosen randomly from 1 to n. We select the learning rate from {5e-05, 1e-05} and use the AdamW optimizer and linear learning rate scheduling with warmup. We use a total batch size of 64 and train for {5000, 10000} steps

with early stopping if metrics stop increasing. For the backbone transformer, we use the pre-trained GPT-2 137M from Hugging Face [2]. The Phi-3-mini [3] was finetuned in the same multi-task setting with AdamW for 15 epochs on the dataset with context size 4096 tokens, learning rate 1e-05 and cosine scheduling with warmup. We used up to 4 Nvidia A100 80Gb per experiment.

		input size												
Model	**#param**	**≤32k**	0K	1K	2K	4K	8K	16K	32K	64K	128K	512K	1M	10M
MTL: one model for all tasks														
GPT-2	137M	6	27	15										
Llama-2-7B-32K-Instruct	7B	39	49	52	49	43	40	35	5					
Mistral-7b-Instruct-v0.2	7B	49	60	56	52	49	45	42	37					
01-ai/Yi-9B-200k	9B	45	52	55	48	46	45	36	37	29	24			
Phi-3-mini-128k-instruct	3.8B	52	64	57	55	51	50	46	42	37	7			
c4ai-command-r-v01	51B	59	64	64	63	61	59	52	51	46	38			
GPT-4		72	87	81	77	74	71	64	53	43	36			
~ Phi-3-mini-128k multi-task	3.8B	85	94	92	87	79	88	82	**72**	**58**	10			
Llama3-ChatQA-1.5-8B + RAG	8B	46	48	48	47	46	45	45	44	42	**45**	**42**	**39**	**37**
~ RMT multi-task	**137M**	75	96	91	83	75	67	61	52	42	35	26	24	24
~ RMT multi-task + question at start	**137M**	**87**	**98**	**96**	**94**	**91**	**89**	**77**	63	37	33	28	23	23
STL: individual model for each task														
~ RMT single-task	137M	91	99	97	95	92	90	86	78	70	59	46	43	34
~ Mamba single-task	130M	99	98	99	99	99	99	99	98	97	93			

Figure 1: Average accuracy on QA1-QA5. RMT trained on BABILong outperforms LLMs in the multi-task setting for task lengths up to 32k tokens. For longer contexts, retrieval-augmented Llama-3 maintains higher performance. The ∼ symbol denotes that the model was finetuned on BABILong. STL stands for single-task learning, where for each task an individual model is trained, and MTL means multi-task learning when one model is trained on a mixture of tasks. $\leq 32k$ is an average score over individual values up to 32k.

Figure 1 displays a performance comparison between language models on BABILong tasks in single-task (STL) and multi-task (MTL) learning settings. We compare the trained models with LLMs on first five tasks to match the evaluation setting in [1]. RMT and Mamba in the bottom section are finetuned in single-task mode, with individual models trained for each task and results averaged across tasks. All other results in the top section were obtained by reusing the same model for all five tasks. This presents a more serious

[2] https://huggingface.co/openai-community/gpt2
[3] https://huggingface.co/microsoft/Phi-3-mini-128k-instruct

challenge, which is represented by the difference in RMT scores.

Finetuning Phi-3-mini yields a substantial average accuracy improvement by more than 30%. However, even after training the model experiences strong performance degradation when the context size approaches its 128k token limit. Despite the complexity of the MTL setting, multi-task RMT achieves the highest scores for sequence lengths up to 32k tokens among LLMs with more than 100 times more parameters. Repeating questions at the beginning of input increases the accuracy by 12% on average. With increasing context size, the performance of multi-task RMT decays faster than in the single-task setting.

V. MULTITASK LEARNING AND GENERALIZATION

In order to better understand the performance of multitask-RMT on each individual task, we study how training on a task affects the performance of the model on other ones, and whether there are benefits compared to single-task learning. In the following two sections, we study the short-context 512-token versions of BABILong. This eliminates the need to filter facts from large amounts of irrelevant text, allowing one to focus more on the tasks themselves.

A. Effect of Training Set Size

The aforementioned results indicate that RMT, when trained on 10 tasks, surpasses the performance of larger LLMs. This raises the question: what is the minimal number of tasks required to achieve such performance? To address this, we incrementally increase the number of tasks in the training set and measure the average performance on QA1-QA10. We choose QA1-QA10 as the evaluation domain, leaving QA11-QA15 out of the training set to measure inductive performance on out-of-domain tasks.

The in-domain evaluation results are depicted in the Figure 2 (left). As expected, the in-domain performance gradually increases with the addition of more training tasks. The inclusion of tasks QA3 and QA6 significantly boosts performance compared to QA2 and QA4. Beyond six tasks, the performance saturates, with only a slight gain from additional tasks. Remarkably, with just six tasks out of ten, RMT already outperforms the few-shot Phi-3-mini model, and with eight tasks surpasses the GPT-4 model.

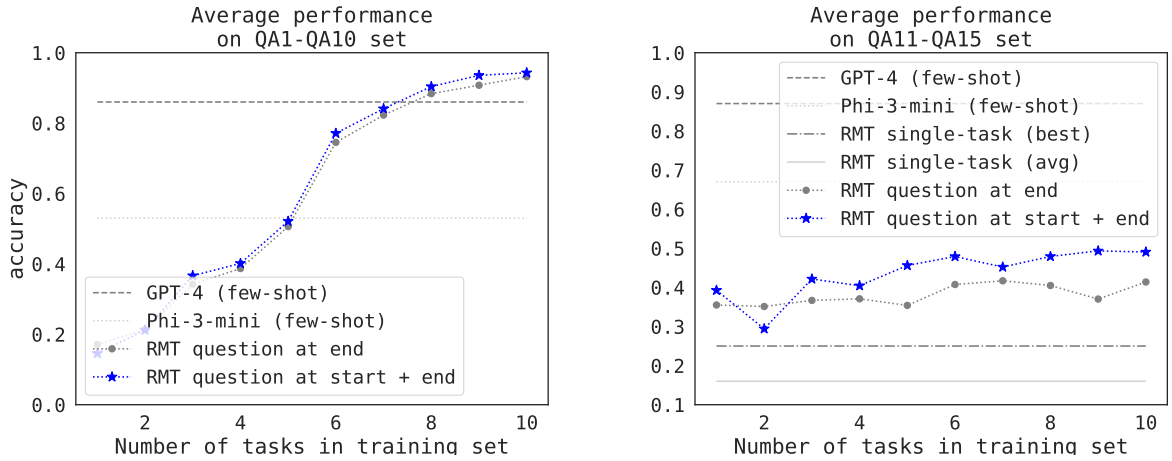

Figure 2: Increasing the number of tasks in the RMT training set improves the multi-task performance on the in-domain tasks QA1-QA10 **(left)** and generalization on unseen tasks QA11-QA15 **(right)**. The task length is set to 512 tokens, which is equal to one segment.

The inductive reasoning capabilities, visualized in Figure 2 (right), show a more modest performance increase, approximately 10% when adding up to 10 tasks. The generalization ability is primarily constrained by the smaller size of the backbone GPT-2 model. Nonetheless, despite underperforming compared to larger models, RMT trained in a multi-task setting significantly outperforms any single-task model. This shows the critical role of multi-task learning in enhancing an agent's generalization abilities.

B. Reciprocal Effect of BABILong Tasks

An important question is, how are the reasoning skills for different tasks related? If one skill is related to another one, then training on the first task will improve the accuracy of the latter. First, we select all 7 BABILong tasks that have the same set of labels: 'bathroom', 'bedroom', 'garden', 'hallway', 'kitchen', and 'office'. This way, by training a model for classification with these labels, the model will likely be able to transfer its knowledge from one task to another.

We start by training a single model on each of the seven tasks and measure its performance on all other six tasks, Figure 3, left. The results indicate that tasks such as QA1 and QA12 are relatively easy and can be partially learned by training on other tasks. On the other hand, QA3 requires precisely tracking multiple objects, and QA4 with spatial relations can be learned only by training on these tasks themselves. Additionally, QA11 and QA13 have

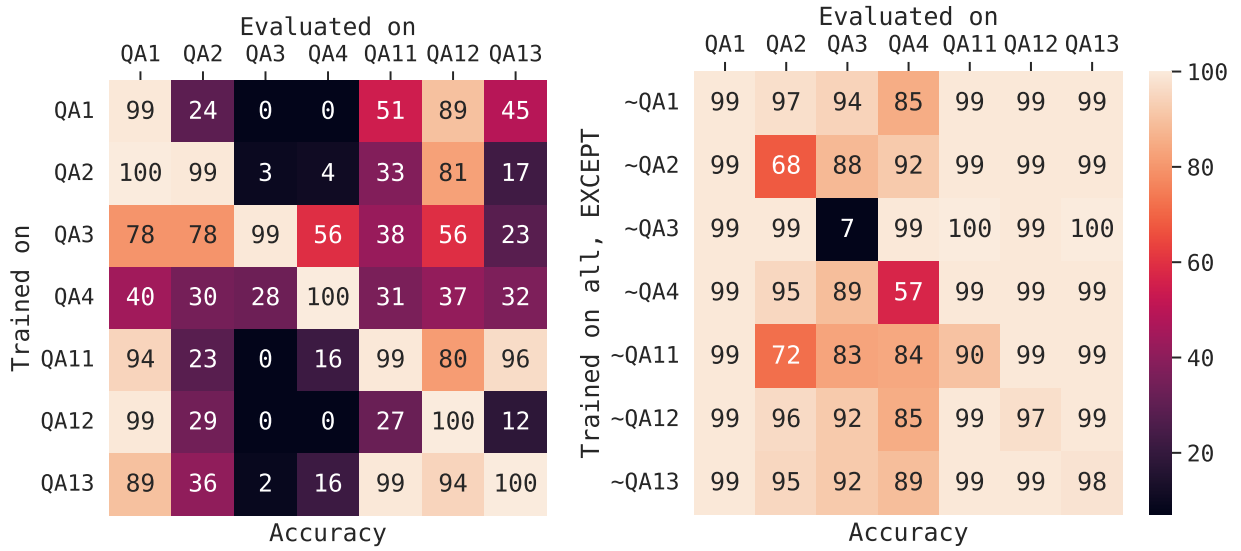

Figure 3: **Left**: Training on BABILong tasks improves the performance on other tasks with the same label. **Right**: Training on other tasks with the same label can teach RMT to solve the selected task with near-perfect accuracy. The task length is set to 512 tokens, equal to one segment.

a positive influence on each other. For QA12 and QA13 the relation is asymmetric, QA13 being helpful for QA12, but not the other way around. Simpler tasks like QA1 and QA12 seem to have near to no effect on complex QA3 and QA4.

The findings are confirmed by the experiment on Figure 3, right. In this setting, we train the model on all six tasks, except the selected one, and perform evaluation on all seven tasks. Similarly to the single-task training, QA3 and QA4 remain complex even when all other skills are learned. On the opposite, the QA1, QA12, and QA13 can be learned nearly perfectly, even if they are not included in the training set. The provided results confirm the diversity and non-trivial structure of BABILong and underscore the necessity of learning more than one skill to solve the whole benchmark successfully.

C. Generalization to context length

As previously shown, the strong feature of RMT is the ability to generalize to unseen sequence lengths on the training task. However, does this ability hold up with unseen tasks? To answer this question we evaluate the performance of multi-task RMT version trained on QA1-QA10 on the out-of-domain QA11-QA15. To compare it with single-task models we calculate the scores of ten models trained on individual tasks from QA1 to QA10 and

average their scores on tasks QA11 to QA15 for each sequence length. All RMT models here are trained on up to 32 segments, which is equivalent to 16k tokens, and evaluated on sequences from 0k to 128k tokens.

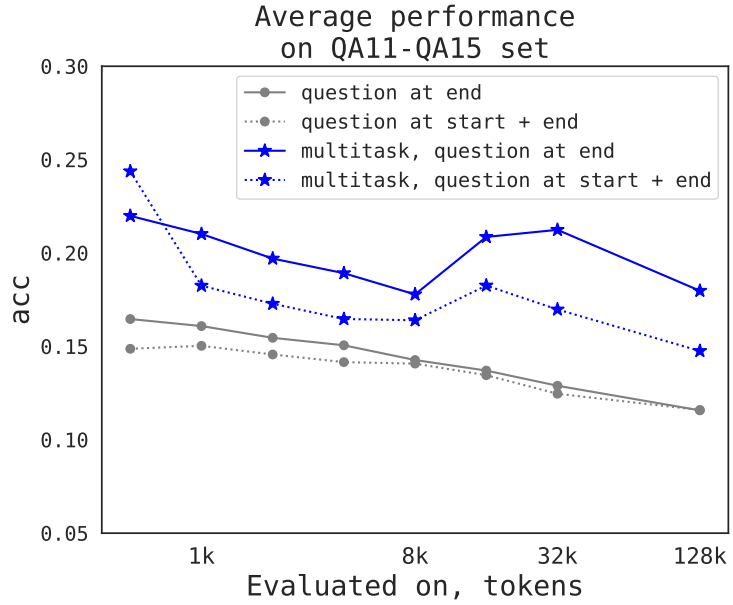

Figure 4: Multitask learning enhances generalization both to new tasks and context sizes. We compare multi-task RMT trained on 32 segments of QA1-QA10 with the average performance of single-task versions trained on individual QA1-QA10 tasks. Each version includes two variations: one with the question at the end, and another with the question at both the beginning and end. Scores are averaged across tasks QA11-QA15.

The results of long-context models on short sequences indicate similar behavior to short-context models: the multi-task RMT significantly outperforms the single-task versions. The multi-task advantage holds up even when the context size is increased. The fact that RMT partially retains its performance on unseen lengths of unseen tasks supports the assumption about generalization skills gained with multitask learning. The plots of multitask models exhibit visible bumps at context sizes 16k and 32k. This can be explained by overfitting on the context size in training, which leads to better performance on this length.

D. Robustness

We hypothesize that multi-task learning has a positive effect not only on generalization but also on robustness for changes in task formulation. To confirm this, we change the

names of people, objects, and places in five tasks QA1-QA5. In order to retain the output distribution for the model, for each task we keep the labels and corresponding entities in input unchanged. RMT versions are evaluated on each task on various sequence lengths from 0k to 32k, and the scores are averaged across five tasks, see Figure 5.

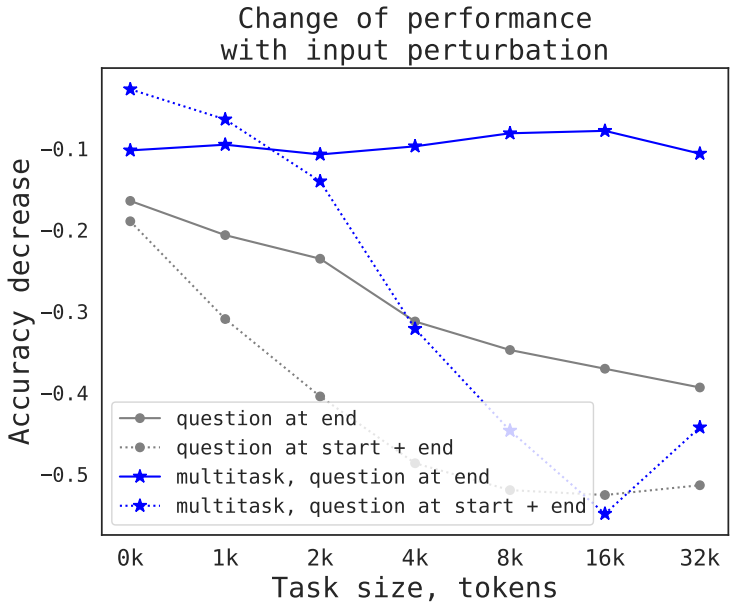

Figure 5: Multi-task learning enhances RMT's robustness to input perturbations. We measure the average performance on QA11-QA15 tasks with changed entity names in input across different sequence lengths. All models are trained on 32 segments; multi-task versions are trained on QA1-QA10 tasks simultaneously, while the single-task plot shows the average performance of models trained individually on QA1-QA10 tasks.

The multi-task versions indeed show an increased robustness that holds up even when the sequence length is increased up to 32k tokens. Interestingly, the RMT version with question repeated at the start and end of the sequence, maintains better performance on shorter sequences but quickly degrades on longer ones. When the question is at the end, the model experiences close to no degradation for all tested sequence lengths. This may be attributed to the necessity for "question in start + end" models to store the question in memory for each segment. With perturbations in the question itself, the scaled task becomes increasingly more complex compared to the training distribution.

VI. DISCUSSION

Our experiments confirm that long-context tasks present significant challenges even for the most powerful language models. By finetuning smaller models, we have demonstrated a substantial performance improvement on BABILong tasks. For RMT, the introduced input arrangement with question, in the beginning, enhances in-domain performance, while placing the question at the end sometimes yields better results for inductive evaluation on unseen tasks and context sizes. This effect might be due to overfitting to question options in the training set, which should be improved with increased task diversity.

The proposed multitask learning procedure consistently boosts stability to input changes and improves generalization to new tasks, lengths, or both. These abilities are likely constrained by the size of the backbone model and the memory state. Selecting larger backbones and extending memory size, for example with the associative memory mechanism, could further enhance performance.

While multitask learning offers significant benefits, training models on single tasks remains a viable option with marginally better performance when generalization is not required. Despite the clear limitations in robustness and out-of-domain performance, this still may be the preferred approach in some cases. Employing a backbone with stronger language abilities can help improve multi-task performance with RMT. Multi-task RMT models exhibit faster degradation on longer contexts compared to single-task RMT models and the RAG pipeline. One of the approaches to mitigate overfitting to context length is by employing more aggressive length sampling during training and experimentation with the curriculum procedure.

RMT's generalization to new tasks is weaker compared to large language models, especially for tasks with unseen labels. This limitation is largely due to the smaller size of the GPT-2 backbone and could be addressed by using a stronger backbone model. Despite these challenges, our approach demonstrates that smaller models can be effectively fine-tuned for complex long-context tasks, eliminating the need to train larger models or prepare a model for each task.

VII. CONCLUSION

In this paper, we address a significant gap in the performance of language models in reasoning with long contexts. We demonstrate that even a small 137M Recurrent Memory Transformer (RMT) model can master multiple BABILong tasks simultaneously, outperforming much larger language models. By proposing an efficient input arrangement and training on multiple tasks, we achieved state-of-the-art results in the multi-task setting of the BABILong QA1-QA5 set for up to 32k tokens.

Our extensive analysis of performance on BABILong tasks highlights the interconnection and diversity of respective long-context reasoning skills. The complexity of generalization to new tasks and sequence lengths highlights the efficiency of BABILong tasks as a proxy for real-world long-context LLM evaluation. By training RMT on multiple tasks, we significantly reduce the computational cost compared to LLMs and eliminate the need for training multiple models for individual tasks. We show that multi-task training not only improves generalization to new tasks but also enhances robustness to changes in input. These advancements underscore the potential of multi-task learning in combination with recurrent models for making long-context processing more accessible and efficient for practical applications.

VIII. FUNDING

This work was supported by a grant for research centers, provided by the Analytical Center for the Government of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000D730324P540002) and the agreement with the Moscow Institute of Physics and Technology dated November 1, 2021 No. 70-2021-00138.

IX. ETHICS APPROVAL AND CONSENT TO PARTICIPATE

This work does not contain any studies involving human and animal subjects.

X. CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

References

1. Y. Kuratov, A. Bulatov, P. Anokhin, I. Rodkin, D. Sorokin, A. Sorokin, and M. Burtsev, *Babilong: Testing the limits of llms with long context reasoning-in-a-haystack* (2024), 2406.10149.
2. A. Bulatov, Y. Kuratov, and M. Burtsev, in *Advances in Neural Information Processing Systems* (2022), vol. 35, pp. 11079–11091.
3. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, in *Advances in neural information processing systems* (2017), pp. 5998–6008, URL `http://papers.nips.cc/paper/7181-attention-is-all-you-need`.
4. OpenAI, *Gpt-4 technical report* (2023), 2303.08774.
5. Anthropic, *Introducing the next generation of claude* (2024), URL `https://www.anthropic.com/news/claude-3-family`.
6. M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser, et al., in *arXiv preprint arXiv:2403.05530* (2024).
7. A. Gu and T. Dao, in *arXiv preprint arXiv:2312.00752* (2023).
8. J. Weston, A. Bordes, S. Chopra, and T. Mikolov, in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, edited by Y. Bengio and Y. LeCun (2016), URL `http://arxiv.org/abs/1502.05698`.
9. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever (2019).
10. I. Beltagy, M. E. Peters, and A. Cohan, in *arXiv preprint arXiv:2004.05150* (2020).
11. M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, et al., in *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Curran Associates, Inc., 2020), vol. 33, pp. 17283–17297, URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf`.
12. J. Ding, S. Ma, L. Dong, X. Zhang, S. Huang, W. Wang, and F. Wei, in *arXiv preprint arXiv:2307.02486* (2023).
13. B. Peng, E. Alcaide, Q. Anthony, A. Albalak, S. Arcadinho, H. Cao, X. Cheng, M. Chung,

M. Grella, K. K. GV, et al., in *arXiv preprint arXiv:2305.13048* (2023).

14. S. De, S. L. Smith, A. Fernando, A. Botev, G. Cristian-Muraru, A. Gu, R. Haroun, L. Berrada, Y. Chen, S. Srinivasan, et al., in *arXiv preprint arXiv:2402.19427* (2024).
15. M. Beck, K. Pöppel, M. Spanring, A. Auer, O. Prudnikova, M. Kopp, G. Klambauer, J. Brandstetter, and S. Hochreiter, in *arXiv preprint arXiv:2405.04517* (2024).
16. A. Gu, K. Goel, and C. Re, in *International Conference on Learning Representations* (2021).
17. Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, Florence, Italy, 2019), pp. 2978–2988, URL `https://aclanthology.org/P19-1285`.
18. Q. Wu, Z. Lan, K. Qian, J. Gu, A. Geramifard, and Z. Yu, in *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022* (Association for Computational Linguistics, Online only, 2022), pp. 308–318, URL `https://aclanthology.org/2022.findings-aacl.29`.
19. W. S. McCulloch and W. Pitts, in *The bulletin of mathematical biophysics* (Springer, 1943), vol. 5, pp. 115–133.
20. C. Stephen, in *Automata studies* (Princeton University Press, 1956).
21. P. J. Werbos, in *Proceedings of the IEEE* (IEEE, 1990), vol. 78, pp. 1550–1560.
22. S. Hochreiter and J. Schmidhuber, in *Neural Comput.* (MIT Press, Cambridge, MA, USA, 1997), vol. 9, p. 1735–1780, ISSN 0899-7667, URL `https://doi.org/10.1162/neco.1997.9.8.1735`.
23. A. Graves, G. Wayne, and I. Danihelka, in *arXiv preprint arXiv:1410.5401* (2014).
24. J. Weston, S. Chopra, and A. Bordes, in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, edited by Y. Bengio and Y. LeCun (2015), URL `http://arxiv.org/abs/1410.3916`.
25. A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, in *Advances in neural information processing systems* (2019), vol. 32.
26. R. Caruana, in *Machine Learning* (1997), vol. 28, pp. 41–75, URL `https://api.semanticscholar.org/CorpusID:45998148`.
27. S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, in *Proceedings of the AAAI conference on artificial intelligence* (2019), vol. 33, pp. 4822–4829.
28. D. Karpov and V. Konovalov, in *Computational Linguistics and Intellectual Technologies*

(2023), vol. 2023.

29. Y. Bai, X. Lv, J. Zhang, Y. He, J. Qi, L. Hou, J. Tang, Y. Dong, and J. Li, in *arXiv preprint arXiv:2401.18058* (2024).
30. T. Li, G. Zhang, Q. D. Do, X. Yue, and W. Chen, in *arXiv preprint arXiv:2404.02060* (2024).
31. C. An, S. Gong, M. Zhong, M. Li, J. Zhang, L. Kong, and X. Qiu, in *arXiv preprint arXiv:2307.11088* (2023).
32. X. Zhang, Y. Chen, S. Hu, Z. Xu, J. Chen, M. K. Hao, X. Han, Z. L. Thai, S. Wang, Z. Liu, et al., in *arXiv preprint arXiv:2402.13718* (2024).
33. C.-P. Hsieh, S. Sun, S. Kriman, S. Acharya, D. Rekesh, F. Jia, and B. Ginsburg, in *arXiv preprint arXiv:2404.06654* (2024).
34. J. W. Rae, A. Potapenko, S. M. Jayakumar, C. Hillier, and T. P. Lillicrap, in *International Conference on Learning Representations* (2020), URL `https://openreview.net/forum?id=SylKikSYDH`.