EXTENDING LLM CONTEXT VIA ASSOCIATIVE RECURRENT MEMORY

Gleb Kuzmin[1,4,7] **Ivan Rodkin**[2,6] **Aydar Bulatov**[3,6] **Yuri Kuratov**[3,6]
Timothy Baldwin[2] **Mikhail Burtsev**[5] **Artem Shelmanov**[2]
[1]FusionBrain Lab, AXXX [2]MBZUAI [3]AXXX [4]RUDN
[5]London Institute for Mathematical Sciences [6]MIRAI
[7]Laboratory for Analysis and Controllable Text Generation Technologies RAS
kuzmin.gyu@gmail.com ivan.rodkin@mbzuai.ac.ae bulatov@cogailab.com
kuratov@cogailab.com timothy.baldwin@mbzuai.ac.ae
mb@lims.ac.uk artem.shelmanov@mbzuai.ac.ae

ABSTRACT

Closed-source LLMs support context windows of up to 1M tokens and beyond, while smaller open-source models still have a limited window of 32K-128K tokens. However, for some tasks, it is necessary to use a small local LLM to avoid any data leaks; some of these tasks require domain-specific knowledge and long-context understanding. To address this gap, we introduce two domain-specific long-context datasets, ManyTypes-long and GovReport-long, and present a practical recipe for extending short-context LLMs using the Associative Recurrent Memory Transformer (ARMT) architecture. Finally, we analyze the associative memory in trained ARMT models and show that associative memory primarily benefits from representations in the middle and upper layers of the transformer, allowing us to reduce the size of the models by removing redundant associative memory from other layers. Our results demonstrate an effective approach to enabling long-context capabilities in small, privacy-preserving LLMs for domain specific tasks.

1 INTRODUCTION

Long-context understanding is crucial for many tasks, such as processing and understanding technical and financial reports, software engineering, and multi-document reasoning in scientific and legal domains. These scenarios often require models to integrate information distributed across thousands or even millions of tokens. However, standard Transformer architectures (Vaswani et al., 2017) struggle to scale to such contexts, as the computational and memory costs of self-attention grow quadratically with sequence length. Moreover, Transformer performance degrades as the context length increases (Liu et al., 2024; Kuratov et al., 2024). Therefore, since the introduction of the Transformer architecture, long-context processing has emerged as a central and rapidly evolving research direction (Beltagy et al., 2020; Katharopoulos et al., 2020; Bulatov et al., 2022). Traditionally, efficient long-context approaches are built using recurrent architectures (Gu & Dao, 2024; Peng et al., 2023); however, such models typically must be trained from scratch, which limits the ability to leverage existing pretrained LLMs. Moreover, fully-recurrent LMs update the memory at each time step, which complicates high-level information processing in tasks such as structured copying (Jelassi et al., 2024) and instruction following (Park et al., 2024).

Recently, Bulatov et al. (2024); Rodkin et al. (2024) proposed augmenting pretrained LLMs with recurrent segment-level Transformer mechanisms designed for long-context processing. These approaches preserve strong intra-segment modeling performance while enabling linear scaling with respect to context length. In this work, we focus on the Associative Recurrent Memory Transformer (ARMT) Rodkin et al. (2024), which introduces a capacious segment-level associative memory and features strong scaling to extremely long input sizes. Prior work on ARMT-based models has been limited to scales below 200M parameters and evaluated on a narrow set of tasks (Rodkin et al., 2024), leaving their behavior at larger model sizes mostly unexplored. Models at this scale typically struggle to handle complex real-world workloads. In this work, we extend ARMT to small-

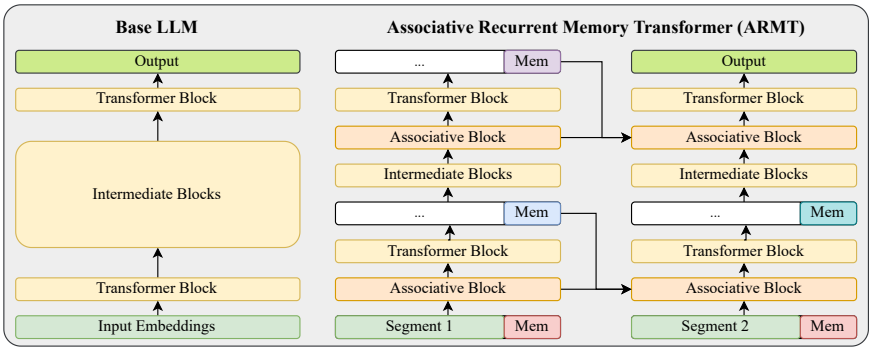

Figure 1: Base LLM architecture (left) and ARMT architecture (right). ARMT divides the input text into segments and processes them sequentially, allowing the model to handle long contexts.

and medium-sized LMs (1–3B parameters), which are substantially more capable in practical settings. These models provide a practical middle ground, enabling linear-compute, constant-memory long-context processing while maintaining strong performance on real-world tasks.

Our contributions are threefold: **(1)** we construct two new datasets with real-world tasks to train and evaluate the long-context performance of LLM, with a focus on narrow domain finetuning; **(2)** we provide a training recipe for the context extension of LLMs with the help of associative memory; **(3)** we analyze which hidden representations of LLM are most important for associative memory, showing that even four associative blocks are enough for efficient context extension compared to the original ARMT design, which includes an associative block in every transformer layer.

2 Background

The overview of the ARMT architecture is presented in Figure 1. ARMT wraps a base backbone LLM and enables segment-wise processing of input segments, splitting long-context inputs into non-overlapping segments of fixed length. Each segment is also augmented with trainable memory embeddings (Mem), which are processed throughout the model. In the core of the ARMT is the Associative Block, which transforms the layer's input with associative memory, using memory and associations from previous segments. The associative memory mechanism consists of three parts:

- **Memory extraction**: each transformer layer compresses an input segment into memory embeddings.
- **Memory consolidation**: memory embeddings are then consolidated in a per-layer associative matrix as key-value pairs.
- **Association**: every embedding of the following segment will be transformed into a query vector and multiplied by this associative matrix.

Appendix A presents the formal definition of the Associative Block.

ARMT can be viewed as combining the best of both worlds: the ability of recurrent architectures to propagate information across arbitrarily long contexts (in principle), and the strong performance of full self-attention within a bounded context window.

3 LLM Context Extension via ARMT

Our recipe for extending the LLM context via ARMT consists of three main ingredients:

Synthetic long context training data. We simulated the original dataset collection procedure, generating the question-answer pairs for several short-context paragraphs from a long context document, after combining it into a long context and using for training. To ensure diversity in the data, we used several models from different model families for generation of QA pairs.

Curriculum learning. In curriculum learning, we varied the complexity of the task as a maximum length of the sequence and, correspondingly, the maximum segments of ARMT, using curricula with 2-4-8 segments of 1024 tokens. We also annealed the learning rate in the stages of the curriculum, gradually reducing it for the later stages.

Sparcifying Associative layers. As an optional step, we propose to remove some associative layers after training, showing that only four layers instead of 26 are enough to achieve similar performance.

4 Long-Context Datasets

We construct long-context datasets for training and testing using the ManyTypes4Py (Mir et al., 2021) and GovReport-QS (Cao & Wang, 2022) datasets.

ManyTypes-long (MT) targets variable type prediction in code with a long context. We split the original dataset into non-overlapping repositories for training, validation, and test splits and stacked the code scripts from each repository to obtain a long text with the desired length of up to 64k tokens. We reused the original labels for the variable types from ManyTypes4Py (Mir et al., 2021).

GovReport-long (GR) focuses on long-document question answering. The original GovReport-QS contains triplets {report, question, answer} with ground-truth paragraphs in the report for each question. We used ground-truth paragraphs from the report and mixed them with non-relevant paragraphs, keeping the paragraph order to build datasets of up to 64k in length. Due to the limited training size of the base GR dataset, we enlarged it with synthetic data to build a GR-100+ dataset while keeping the validation and test sets unchanged to avoid distribution shifts.

For both datasets, to avoid the model's answers being based on parametric knowledge, we placed the question before the context in our prompts. The examples from the MT and GR datasets are presented in Tables 6 and 7 in Appendix B.1. Dataset statistics are provided in Table 5 in Appendix B.1.

5 Experiments

5.1 Experimental Setup

Base LLM. We experimented with Gemma-3-1B-IT (Team et al., 2025) as a base LLM. This model was chosen because it is one of the state-of-the-art LLMs in its category for short-context benchmarks, but it has a relatively short maximum length of 32k and significant performance degradation in long-context scenarios (Team et al., 2025), even within this context window. We hypothesize that this could arise from the use of interleaving sliding-window attention in its architecture instead of full attention. Therefore, this model is a good candidate for context extension.

Evaluation and metrics. We fine-tune the models on the training sets and evaluate the performance on the test sets of MT and GR. We used the exact match (EM) metric for the MT dataset, as the target answers are short and require precise variable type prediction. For the GR and GR-100+ datasets, we used the ROUGE-L (Lin, 2004) metric because the ground truth answers in GR are free-form and longer than those in MT.

Training details. We fine-tuned LLMs using LoRA Hu et al. (2021) for GPU memory efficiency. We used the training setup from Section 3 for both GR and MT datasets. We also experimented with additional pretraining for the ARMT model on a synthetic long-context QA task; however, this pretraining limits the generalization capability of ARMT. The results of the pretraining experiments are presented in Appendix C.3. Training hyperparameters are provided in Appendix D.

5.2 Results

Main results. The results on the MT dataset are presented in Table 1. The base model without fine-tuning exhibits near-zero performance across all context lengths, highlighting the necessity of specialized long-context adaptation. Fine-tuning the base model on the 8k dataset (Base, MT, 8k) yields stable performance up to its maximum supported length of 32k; however, performance drops substantially beyond this limit. In contrast, the ARMT-augmented model demonstrates consistent

Table 1: Best results on the MT dataset for Gemma-3-1B-IT model, metric - EM. ARMT shows comparable with the base model overall performance, and outperforms it on OOD and Long-OOD.

Model/ Lengths	No Fine-Tuning	Base, MT, 8k	ARMT, MT, 2k	ARMT, MT, 4k	ARMT, MT, 8k
0k-1k	0.000	0.795	0.782	0.782	0.756
1k-2k	0.000	0.790	0.724	0.714	0.743
2k-4k	0.000	0.774	0.689	0.698	0.736
4k-6k	0.000	0.767	0.616	0.644	0.699
6k-8k	0.000	0.859	0.609	0.717	0.804
8k-10k	0.000	0.852	0.506	0.753	0.753
10k-12k	0.011	0.868	0.374	0.670	0.791
12k-14k	0.000	0.787	0.404	0.702	0.745
14k-16k	0.000	0.764	0.337	0.640	0.685
16k-24k	0.000	0.812	0.229	0.646	0.757
24k-32k	0.000	0.713	0.198	0.584	0.713
32k-49k	0.000	0.591	0.139	0.609	0.765
49k-65k	0.000	0.317	0.089	0.604	0.713
In-Domain (0k-8k)	0.000	0.797	0.685	0.711	0.749
OOD (8k-65k)	0.001	0.709	0.271	0.647	0.741
Long-OOD (32k-65k)	0.000	0.463	0.116	0.607	0.741
Full (0k-65k)	0.001	0.741	0.419	0.670	0.744

Table 2: Best results on the GR-100+ dataset for Gemma-3-1B-IT model, ROUGE-L. ARMT shows slightly lower overall performance than the base model, but outperforms it on Long-OOD.

Model/ Lengths	No Fine-, Tuning	Base, GR-100+, 8k	ARMT, GR-100+, 2k	ARMT, GR-100+, 4k	ARMT, GR-100+, 8k
0k-1k	0.188	0.380	0.357	0.357	0.358
1k-2k	0.161	0.385	0.296	0.294	0.303
2k-4k	0.140	0.352	0.264	0.275	0.279
4k-6k	0.114	0.344	0.240	0.286	0.303
6k-8k	0.113	0.296	0.219	0.238	0.254
8k-10k	0.106	0.306	0.205	0.233	0.235
10k-12k	0.101	0.269	0.213	0.221	0.241
12k-14k	0.091	0.280	0.193	0.225	0.237
14k-16k	0.088	0.231	0.180	0.217	0.240
16k-24k	0.083	0.269	0.115	0.200	0.252
24k-32k	0.078	0.247	0.016	0.168	0.207
32k-49k	0.078	0.171	0.002	0.128	0.200
49k-65k	0.103	0.189	0.000	0.040	0.264
In-Domain (0k-8k)	0.143	0.351	0.275	0.290	0.299
OOD (8k-65k)	0.092	0.266	0.162	0.211	0.238
Long-OOD (32k-65k)	0.084	0.175	0.002	0.108	0.215
Full (0k-65k)	0.116	0.306	0.215	0.248	0.266

improvements throughout curriculum learning. Performance steadily increases across training stages (2k, 4k, and 8k). The final ARMT model (ARMT MT 8k) achieves comparable average performance within the training regime while substantially improving generalization to out-of-domain context lengths, effectively extending the base model's capability to handle longer inputs.

The results on GovReport-long are presented in Table 2. Because the base version of the GR dataset is relatively small, we report results on this dataset separately in Table 12 (Appendix C.2) and focus our analysis on the enlarged GR-100+ dataset. In zero-shot evaluation, the base model achieves low performance, with an average ROUGE-L score of 0.116. After fine-tuning on GR-100+ (Base, GR-100+, 8k), the model reaches an average ROUGE-L of 0.306, but its performance on Long-OOD examples drops substantially to 0.175 ROUGE-L. The ARMT-augmented model (ARMT, GR-100+, 8k) partially mitigates this degradation, improving ROUGE-L on Long-OOD to 0.215, although performance on shorter context lengths becomes slightly lower.

Overall, the results provide consistent evidence that the proposed recipe effectively extends the usable context length of Transformer-based LLMs.

Associative layers ablation. Integrating associative memory layers into a pre-trained LLM alters its internal representations, making subsequent fine-tuning necessary to restore performance. Moreover, the extent of architectural modification directly affects the amount of computation required for re-adaptation. This motivates the following research question: *what is the minimal architectural transformation needed for effective associative memory integration, and which layers are most important for inserting associative memory modules?* To investigate this, we introduce associative

Table 3: Associative layers ablation on the MT dataset for Gemma-3-1B-IT model, metric - EM. Middle and upper layers representations are the most important for associative memory; ARMT with only top-4 associative blocks keeps almost the same performance as the full ARMT.

Model/ Lengths	**Base, MT, 8k**	**ARMT, MT, 8k**	**ARMT, MT, 8k, w/o layers 0-6**	**ARMT, MT, 8k, w/o layers 7-12**	**ARMT, MT, 8k, w/o layers 13-18**	**ARMT, MT, 8k, w/o layers 19-25**	**ARMT, MT, 8k, only top-1 layer**	**ARMT, MT, 8k, only top-4 layers**
0k-1k	0.795	0.756	0.756	0.756	0.756	0.756	0.756	0.756
1k-2k	0.790	0.743	0.733	0.743	0.571	0.676	0.590	0.724
2k-4k	0.774	0.736	0.736	0.726	0.443	0.717	0.594	0.708
4k-6k	0.767	0.699	0.699	0.699	0.329	0.671	0.521	0.712
6k-8k	0.859	0.804	0.815	0.804	0.315	0.739	0.598	0.793
8k-10k	0.852	0.753	0.753	0.765	0.395	0.716	0.531	0.765
10k-12k	0.868	0.791	0.780	0.791	0.341	0.736	0.505	0.758
12k-14k	0.787	0.745	0.766	0.766	0.277	0.809	0.457	0.755
14k-16k	0.764	0.685	0.674	0.674	0.337	0.708	0.472	0.663
16k-24k	0.812	0.757	0.750	0.764	0.306	0.722	0.458	0.729
24k-32k	0.713	0.713	0.703	0.703	0.287	0.683	0.436	0.693
32k-49k	0.591	0.765	0.757	0.765	0.226	0.739	0.504	0.765
49k-65k	0.317	0.713	0.723	0.723	0.356	0.683	0.505	0.713
In-Domain (0k-8k)	0.797	0.749	0.749	0.747	0.482	0.711	0.610	0.738
OOD (8k-65k)	0.709	0.741	0.739	0.745	0.311	0.724	0.481	0.730
Long-OOD (32k-65k)	0.463	0.741	0.741	0.745	0.287	0.713	0.504	0.741
Full (0k-65k)	0.741	0.744	0.743	0.746	0.372	0.720	0.527	0.733

Table 4: Associative layers ablation on the GR-100+ dataset for Gemma-3-1B-IT model, metric - ROUGE-L. Middle and upper layers representations are the most important for associative memory; ARMT with only top-4 associative blocks keeps almost the same performance as the full ARMT.

Model/ Lengths	**Base, GR-100+, 8k**	**ARMT, GR-100+, 8k**	**ARMT, GR-100+, 8k, w/o layers 0-6**	**ARMT, GR-100+, 8k, w/o layers 7-12**	**ARMT, GR-100+, 8k, w/o layers 13-18**	**ARMT, GR-100+, 8k, w/o layers 19-25**	**ARMT, GR-100+, 8k, only top-1 layer**	**ARMT, GR-100+, 8k, only top-4 layers**
0k-1k	0.380	0.358	0.358	0.358	0.358	0.358	0.358	0.358
1k-2k	0.385	0.303	0.306	0.301	0.232	0.234	0.207	0.299
2k-4k	0.352	0.279	0.277	0.279	0.186	0.212	0.180	0.285
4k-6k	0.344	0.303	0.295	0.273	0.165	0.198	0.160	0.278
6k-8k	0.296	0.254	0.246	0.243	0.165	0.188	0.144	0.244
8k-10k	0.306	0.235	0.238	0.219	0.155	0.190	0.133	0.233
10k-12k	0.269	0.241	0.237	0.226	0.152	0.188	0.129	0.241
12k-14k	0.280	0.237	0.233	0.223	0.156	0.176	0.146	0.230
14k-16k	0.231	0.240	0.234	0.222	0.154	0.169	0.140	0.237
16k-24k	0.269	0.252	0.248	0.220	0.155	0.182	0.121	0.247
24k-32k	0.247	0.207	0.212	0.206	0.140	0.187	0.129	0.217
32k-49k	0.171	0.200	0.213	0.220	0.152	0.160	0.147	0.229
49k-65k	0.189	0.264	0.327	0.286	0.233	0.205	0.207	0.331
In-Domain (0k-8k)	0.351	0.299	0.296	0.291	0.220	0.237	0.209	0.293
OOD (8k-65k)	0.266	0.238	0.237	0.222	0.154	0.181	0.135	0.237
Long-OOD (32k-65k)	0.175	0.215	0.239	0.235	0.171	0.170	0.161	0.253
Full (0k-65k)	0.306	0.266	0.264	0.254	0.185	0.207	0.170	0.263

memory layers only into a subset of transformer layers. Concretely, we replace selected associative memory layers with identity mappings after training full ARMT while keeping the remaining blocks unchanged. The results are provided in Tables 3 and 4.

Across datasets, the most important layers are concentrated between layers 13 and 18 (the third quarter of the network), while early layers (0–6) contribute the least and can be removed after training with no measurable performance degradation. A more fine-grained layer-wise ablation shows that only a small subset of layers is critical for performance. In particular, layer 14 consistently emerges as the most important for both tasks, with several additional layers contributing to GR. We therefore evaluate ARMT variants that retain only the top-1 and top-4 associative layers ranked by importance. The results demonstrate that the remaining associative layers can be removed with minimal performance loss, reducing the number of additional layers by a factor of six. Detailed layer-wise ablation results are provided in Tables 8 and 9 in Appendix C.1.

Other ablations. We conducted continuous pretraining for the ARMT model before training on GR-100+; the details are shown in Table 13 (Appendix C.3). This setup achieve higher performance on in-domain lengths compared to the non-pretrained ARMT, the performance on long-context samples is worse. We hypothesize that this performance drop could arise from the limited length of texts in pretraining. However, this setup shows promising results and is left for future research.

6 CONCLUSION

We proposed a recipe for the context extension of LLMs with the ARMT on domain-specific tasks, closing the gap for the critical privacy-preserving long-context understanding with small local models. We introduced two datasets for training and testing of LLMs in long-context tasks: ManyTypes-long and GovReport-long. Finally, we analyzed the contribution of the hidden representation in the associative memory in the ARMT, showing that only the few most important layers are needed to achieve full-model performance.

ACKNOWLEDGEMENTS

We thank anonymous reviewers for their insightful feedback towards improving this paper. This work is supported by a grant #848011 from the MBZUAI & WIS Collaborative Research Program.

REFERENCES

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. Recurrent memory transformer. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 11079–11091. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/47e288629a6996a17ce50b90a056a0e1-Paper-Conference.pdf.

Aydar Bulatov, Yuri Kuratov, Yermek Kapushev, and Mikhail Burtsev. Beyond attention: Breaking the limits of transformer context length with recurrent memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17700–17708, 2024.

Shuyang Cao and Lu Wang. Hibrids: Attention with hierarchical biases for structure-aware long document summarization, 2022.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

Samy Jelassi, David Brandfonbrener, Sham M Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. In *International Conference on Machine Learning*, pp. 21502–21521. PMLR, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention, 2020.

Yury Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *Advances in Neural Information Processing Systems*, 37:106519–106554, 2024.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

A. M. Mir, E. Latoskinas, and G. Gousios. Manytypes4py: A benchmark python dataset for machine learning-based type inference. In *IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*, pp. 585–589. IEEE Computer Society, May 2021. doi: 10.1109/MSR52588.2021.00079.

Jongho Park, Jaeseung Park, Zheyang Xiong, Nayoung Lee, Jaewoong Cho, Samet Oymak, Kangwook Lee, and Dimitris Papailiopoulos. Can mamba learn how to learn? a comparative study on in-context learning tasks. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=GbFluKMmtE.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. RWKV: Reinventing RNNs for the transformer era. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14048–14077, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.936. URL https://aclanthology.org/2023.findings-emnlp.936/.

Ivan Rodkin, Yuri Kuratov, Aydar Bulatov, and Mikhail Burtsev. Associative recurrent memory transformer. *arXiv preprint arXiv:2407.04841*, 2024.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

A Formal Description of the ARMT Architecture

For the hidden states of segment s in layer l H_s^l, memory tokens from the previous layer M_s^{l-1}, the associative matrix A_s^l, and normalization z_s^l are updated as follows:

$$M_s^l = \{m_i\}, \quad M_s^l = \text{TransLayer}(H_s^{l-1}, M_s^{l-1}) \tag{1}$$

$$k_i, v_i = W_K m_i, W_V m_i; \quad \beta_i = \sigma(W_\beta m_i); \quad A_0^l = \vec{0}; \quad z_0^l = \vec{0}; \tag{2}$$

$$\overline{v}_i = \frac{A_{s-1}^l \phi(k_i)}{(z_{s-1})^T \phi(k_i)}; \quad \gamma_i = 1 - \frac{(z_{s-1})^T \phi(k_i)}{\|\phi(k_i)\|^2}; \tag{3}$$

$$A_s^l = A_{s-1}^l + \sum_i \beta_i (v_i - \overline{v}_i) \otimes \phi(k_i); \quad z_s^l = z_{s-1}^l + \sum_i \gamma_i \phi(k_i). \tag{4}$$

Reading from memory in the following segments for embedding x_j from H_{s+1}^{l-1}:

$$q_j = W_Q x_j; \quad y_j = \frac{A_s^l \phi(q_j)}{(z_s^l)^T \phi(q_j)}, \tag{5}$$

where y_j is an association for x_j.

B Datasets Statistics and Ablations

B.1 Dataset Statistics and Examples

In this section, we present statistics for all dataset versions and splits used in our experiments. The dataset overview is summarized in Table 5. We also provide short illustrative examples from the MT and GR datasets in Tables 6 and 7.

Table 5: Dataset statistics - number of samples for each dataset and for each split.

Dataset/ Split	**Train**	**Validation**	**Test**
MT, 2k	229.4k	3.1k	1.3k
MT, 4k	289.2k	4.5k	1.3k
MT, 8k	328.3k	5.1k	1.3k
GR, 2k	9.9k	0.5k	1.0k
GR, 4k	17.4k	0.9k	1.0k
GR, 8k	19.7k	1.0k	1.0k
GR-100+, 2k	90.5k	0.5k	1.0k
GR-100+, 4k	121.7k	0.9k	1.0k
GR-100+, 8k	128.4k	1.0k	1.0k

Table 6: Example from MT dataset. The question during training and evaluation is placed before context (i. e. at the start of the prompt).

Example	**Text**
Context	from typing import Any import typing [docstring] from alembic import op import sqlalchemy as sa from sqlalchemy . ext . declarative import declarative_base from sqlalchemy . orm import sessionmaker , relationship [comment] revision = [string] down_revision = [string] Base = declarative_base ( ) db = sa db . Model = Base db . relationship = relationship def create_session ( ) : connection = op . get_bind ( ) session_maker = sa . orm . sessionmaker ( ) session = session_maker (bind = connection) db . session = session def upgrade ( ) : create_session ( ) [comment] op . alter_column ([string] , [string] , type_ = sa . Text , existing_type = sa . String) [comment] def downgrade ( ) : create_session ( ) [comment] op . alter_column ([string] , [string] , type_ = sa . String , existing_type = sa . Text) [comment] [comment]
Question	What is the type of variable down_revision?
Answer	builtins.str

Table 7: Example from GR dataset. The question during training and evaluation is placed before context (i. e. at the start of the prompt).

Example	**Text**
Context	Economic Significance of Intellectual Property Protection and Theft As we reported in April 2010, IP is an important component of the U.S. economy and IP-related industries pay higher wages and contribute a significant percentage to the U.S. economy. However, the U.S. economy as a whole may grow at a slower pace than it otherwise would because of counterfeiting and piracy's effect on U.S. industries, government, and consumers. Quantifying Economic Impacts Is Difficult, However Industry Research Suggests the Impacts Are Sizable Generally, as we reported in April 2010, the illicit nature of counterfeiting and piracy makes estimating the economic impact of IP infringements extremely difficult, so assumptions must be used to offset the lack of data. Efforts to estimate losses involve assumptions such as the rate at which consumers would substitute counterfeit for legitimate products, which can have enormous impacts on the resulting estimates. Because of the significant differences in types of counterfeited and pirated goods and industries involved, no single method can be used to develop estimates. Each method has limitations, and most experts observed that it is difficult, if not impossible, to quantify the economy-wide impacts. Nonetheless, research in specific industries suggests that the problem is sizeable.
Question	What makes cost-estimates of IP infringements difficult to calculate?
Answer	Generally, as GAO reported in April 2010, the illicit nature of counterfeiting and piracy makes estimating the economic impact of IP infringements extremely difficult.

C Additional Experimental Results

C.1 Ablation Study for Associative Memory Layers

We conducted a layer-wise ablation for associative layers in a trained ARMT model for the MT and GR-100+ datasets. The results are presented in Tables 8 to 11. For the GR-100+ dataset, the most important representations are from the 14-th, 25-th, 9-th, and 13-th layers, while other layers contribute significantly less. For the MT dataset, the most important representations are from the 14-th, 18-th, 25-th, and 19-th layers. Layers are numbered from the starting embeddings layer (i. e., the 0-th layer goes after the embeddings layer, while the 25-th layer goes right before the last layer). We suppose that the middle layers are the most important for the associative memory, as these layers contain the more abstract representation than the lower or higher layers. On the lower layers these representations are still not formed, while on higher ones they are close to the target tokens.

Table 8: Ablation for all associative layers on the GovReport-100+ dataset for ARMT with Gemma-3-1B-IT model, metric - ROUGE-L. Ablated associative layers from 0 to 12.

Model/ Lengths	Base, GR-100+, 8k	ARMT, GR-100+, 8k	W/o layer 0	W/o layer 1	W/o layer 2	W/o layer 3	W/o layer 4	W/o layer 5	W/o layer 6	W/o layer 7	W/o layer 8	W/o layer 9	W/o layer 10	W/o layer 11	W/o layer 12
0k-1k	0.380	0.358	0.358	0.358	0.358	0.358	0.358	0.358	0.358	0.358	0.358	0.358	0.358	0.358	0.358
1k-2k	0.385	0.303	0.307	0.300	0.295	0.302	0.296	0.299	0.303	0.294	0.303	0.290	0.302	0.303	0.302
2k-4k	0.352	0.279	0.278	0.274	0.279	0.280	0.283	0.283	0.279	0.278	0.277	0.269	0.282	0.280	0.296
4k-6k	0.344	0.303	0.298	0.299	0.298	0.297	0.297	0.298	0.301	0.292	0.302	0.280	0.299	0.301	0.285
6k-8k	0.296	0.254	0.247	0.253	0.255	0.246	0.255	0.256	0.254	0.246	0.249	0.240	0.252	0.254	0.254
8k-10k	0.306	0.235	0.228	0.232	0.234	0.239	0.243	0.233	0.237	0.243	0.237	0.211	0.236	0.234	0.231
10k-12k	0.269	0.241	0.231	0.236	0.241	0.244	0.240	0.239	0.244	0.240	0.241	0.227	0.239	0.242	0.242
12k-14k	0.280	0.237	0.238	0.237	0.234	0.230	0.233	0.232	0.235	0.237	0.231	0.216	0.234	0.235	0.227
14k-16k	0.231	0.240	0.238	0.242	0.236	0.238	0.239	0.240	0.240	0.240	0.243	0.218	0.240	0.240	0.239
16k-24k	0.269	0.252	0.250	0.248	0.254	0.257	0.253	0.258	0.249	0.249	0.257	0.228	0.254	0.251	0.246
24k-32k	0.247	0.207	0.212	0.203	0.205	0.214	0.204	0.210	0.204	0.205	0.216	0.190	0.206	0.204	0.216
32k-49k	0.171	0.200	0.199	0.203	0.206	0.214	0.205	0.211	0.204	0.216	0.223	0.170	0.198	0.204	0.226
49k-65k	0.189	0.264	0.271	0.344	0.278	0.284	0.272	0.310	0.264	0.282	0.378	0.170	0.264	0.264	0.284
In-Domain (0k-8k)	0.351	0.299	0.297	0.296	0.297	0.296	0.297	0.298	0.299	0.293	0.297	0.287	0.298	0.299	0.299
OOD (8k-65k)	0.266	0.238	0.235	0.237	0.237	0.239	0.238	0.238	0.238	0.239	0.241	0.216	0.237	0.237	0.236
Long-OOD (32k-65k)	0.175	0.215	0.216	0.236	0.223	0.230	0.220	0.234	0.218	0.231	0.259	0.170	0.213	0.218	0.239
Full (0k-65k)	0.306	0.266	0.264	0.265	0.265	0.266	0.266	0.266	0.266	0.264	0.267	0.249	0.266	0.266	0.265

Table 9: Ablation for all associative layers on the GovReport-100+ dataset for ARMT with Gemma-3-1B-IT model, metric - ROUGE-L. Ablated associative layers from 13 to 26.

Model/ Lengths	Base, GR-100+, 8k	ARMT, GR-100+, 8k	W/o layer 13	W/o layer 14	W/o layer 15	W/o layer 16	W/o layer 17	W/o layer 18	W/o layer 19	W/o layer 20	W/o layer 21	W/o layer 22	W/o layer 23	W/o layer 24	W/o layer 25
0k-1k	0.380	0.358	0.358	0.358	0.358	0.358	0.358	0.358	0.358	0.358	0.358	0.358	0.358	0.358	0.358
1k-2k	0.385	0.303	0.301	0.239	0.304	0.303	0.303	0.289	0.291	0.303	0.303	0.303	0.303	0.296	0.226
2k-4k	0.352	0.279	0.277	0.219	0.279	0.279	0.279	0.274	0.278	0.279	0.279	0.279	0.279	0.279	0.208
4k-6k	0.344	0.303	0.270	0.197	0.302	0.303	0.303	0.287	0.297	0.303	0.303	0.303	0.303	0.301	0.190
6k-8k	0.296	0.254	0.239	0.184	0.254	0.253	0.254	0.263	0.253	0.254	0.254	0.254	0.254	0.253	0.196
8k-10k	0.306	0.235	0.222	0.176	0.235	0.241	0.235	0.232	0.241	0.235	0.235	0.235	0.235	0.234	0.188
10k-12k	0.269	0.241	0.224	0.179	0.242	0.239	0.241	0.232	0.237	0.241	0.241	0.241	0.241	0.241	0.186
12k-14k	0.280	0.237	0.215	0.176	0.237	0.237	0.237	0.234	0.236	0.237	0.238	0.237	0.237	0.235	0.174
14k-16k	0.231	0.240	0.213	0.177	0.241	0.241	0.240	0.233	0.242	0.240	0.239	0.240	0.240	0.243	0.171
16k-24k	0.269	0.252	0.211	0.177	0.252	0.251	0.252	0.249	0.252	0.252	0.252	0.252	0.252	0.251	0.183
24k-32k	0.247	0.207	0.209	0.163	0.207	0.207	0.207	0.209	0.205	0.207	0.207	0.207	0.207	0.208	0.190
32k-49k	0.171	0.200	0.189	0.149	0.200	0.200	0.200	0.222	0.202	0.200	0.200	0.200	0.203	0.203	0.163
49k-65k	0.189	0.264	0.230	0.239	0.264	0.264	0.264	0.273	0.264	0.264	0.264	0.264	0.264	0.264	0.205
In-Domain (0k-8k)	0.351	0.299	0.289	0.239	0.299	0.299	0.299	0.294	0.295	0.299	0.299	0.299	0.299	0.297	0.235
OOD (8k-65k)	0.266	0.238	0.215	0.176	0.238	0.238	0.238	0.234	0.238	0.238	0.238	0.238	0.238	0.238	0.180
Long-OOD (32k-65k)	0.175	0.215	0.198	0.170	0.215	0.215	0.215	0.234	0.216	0.215	0.215	0.215	0.217	0.217	0.173
Full (0k-65k)	0.306	0.266	0.250	0.205	0.267	0.267	0.266	0.262	0.265	0.266	0.266	0.266	0.266	0.265	0.206

Table 10: Ablation for all associative layers on the MT dataset for ARMT with Gemma-3-1B-IT model, metric - EM. Ablated associative layers from 0 to 12.

Model/ Lengths	**Base, MT, 8k**	**ARMT, MT, 8k**	**W/o layer 0**	**W/o layer 1**	**W/o layer 2**	**W/o layer 3**	**W/o layer 4**	**W/o layer 5**	**W/o layer 6**	**W/o layer 7**	**W/o layer 8**	**W/o layer 9**	**W/o layer 10**	**W/o layer 11**	**W/o layer 12**
0k-1k	0.795	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756
1k-2k	0.790	0.743	0.733	0.733	0.733	0.733	0.733	0.733	0.743	0.743	0.743	0.733	0.743	0.743	0.733
2k-4k	0.774	0.736	0.736	0.736	0.736	0.736	0.736	0.736	0.736	0.736	0.726	0.736	0.736	0.736	0.736
4k-6k	0.767	0.699	0.699	0.699	0.685	0.699	0.699	0.699	0.699	0.699	0.699	0.685	0.699	0.699	0.699
6k-8k	0.859	0.804	0.804	0.804	0.815	0.815	0.804	0.815	0.804	0.804	0.815	0.804	0.804	0.804	0.804
8k-10k	0.852	0.753	0.753	0.753	0.741	0.765	0.741	0.728	0.753	0.753	0.778	0.753	0.741	0.753	0.741
10k-12k	0.868	0.791	0.791	0.769	0.791	0.802	0.791	0.802	0.791	0.791	0.813	0.791	0.791	0.791	0.791
12k-14k	0.787	0.745	0.734	0.755	0.734	0.723	0.734	0.734	0.734	0.745	0.766	0.745	0.745	0.745	0.745
14k-16k	0.764	0.685	0.685	0.697	0.685	0.674	0.685	0.685	0.685	0.685	0.674	0.685	0.685	0.685	0.685
16k-24k	0.812	0.757	0.764	0.764	0.764	0.750	0.764	0.757	0.764	0.757	0.764	0.771	0.764	0.757	0.764
24k-32k	0.713	0.713	0.713	0.723	0.703	0.713	0.713	0.703	0.703	0.703	0.693	0.713	0.713	0.713	0.703
32k-49k	0.591	0.765	0.757	0.765	0.757	0.757	0.757	0.757	0.757	0.757	0.757	0.765	0.757	0.765	0.757
49k-65k	0.317	0.713	0.713	0.723	0.713	0.733	0.723	0.693	0.713	0.713	0.743	0.713	0.713	0.713	0.713
In-Domain (0k-8k)	0.797	0.749	0.747	0.747	0.747	0.749	0.747	0.749	0.749	0.749	0.749	0.744	0.749	0.749	0.747
OOD (8k-65k)	0.709	0.741	0.740	0.745	0.738	0.740	0.740	0.734	0.739	0.739	0.749	0.744	0.740	0.741	0.739
Long-OOD (32k-65k)	0.463	0.741	0.736	0.745	0.736	0.746	0.741	0.727	0.736	0.736	0.750	0.741	0.736	0.741	0.736
Full (0k-65k)	0.741	0.744	0.743	0.746	0.741	0.743	0.743	0.739	0.743	0.743	0.749	0.744	0.743	0.744	0.742

Table 11: Ablation for all associative layers on the MT dataset for ARMT with Gemma-3-1B-IT model, metric - EM. Ablated associative layers from 13 to 26.

Model/ Lengths	**Base, MT, 8k**	**ARMT, MT, 8k**	**W/o layer 13**	**W/o layer 14**	**W/o layer 15**	**W/o layer 16**	**W/o layer 17**	**W/o layer 18**	**W/o layer 19**	**W/o layer 20**	**W/o layer 21**	**W/o layer 22**	**W/o layer 23**	**W/o layer 24**	**W/o layer 25**
0k-1k	0.795	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756	0.756
1k-2k	0.790	0.743	0.733	0.610	0.743	0.743	0.743	0.695	0.695	0.743	0.743	0.743	0.752	0.743	0.733
2k-4k	0.774	0.736	0.736	0.491	0.736	0.736	0.736	0.726	0.736	0.736	0.736	0.736	0.726	0.736	0.726
4k-6k	0.767	0.699	0.699	0.384	0.699	0.699	0.699	0.671	0.712	0.699	0.699	0.699	0.685	0.699	0.671
6k-8k	0.859	0.804	0.815	0.370	0.804	0.804	0.804	0.772	0.793	0.804	0.804	0.804	0.793	0.804	0.783
8k-10k	0.852	0.753	0.741	0.407	0.753	0.753	0.753	0.716	0.765	0.753	0.753	0.753	0.765	0.753	0.741
10k-12k	0.868	0.791	0.791	0.330	0.791	0.791	0.791	0.736	0.780	0.791	0.791	0.791	0.769	0.769	0.769
12k-14k	0.787	0.745	0.734	0.255	0.745	0.745	0.745	0.660	0.755	0.745	0.745	0.745	0.755	0.755	0.734
14k-16k	0.764	0.685	0.685	0.360	0.685	0.685	0.685	0.618	0.697	0.685	0.685	0.685	0.719	0.697	0.697
16k-24k	0.812	0.757	0.757	0.333	0.757	0.757	0.757	0.715	0.743	0.757	0.764	0.757	0.764	0.764	0.750
24k-32k	0.713	0.713	0.703	0.287	0.713	0.713	0.713	0.634	0.703	0.713	0.703	0.713	0.703	0.713	0.723
32k-49k	0.591	0.765	0.757	0.278	0.765	0.765	0.757	0.713	0.765	0.765	0.765	0.765	0.765	0.748	0.765
49k-65k	0.317	0.713	0.713	0.356	0.713	0.713	0.713	0.683	0.713	0.713	0.713	0.713	0.723	0.713	0.733
In-Domain (0k-8k)	0.797	0.749	0.749	0.522	0.749	0.749	0.749	0.724	0.738	0.749	0.749	0.749	0.744	0.749	0.735
OOD (8k-65k)	0.709	0.741	0.737	0.323	0.741	0.741	0.740	0.686	0.740	0.741	0.741	0.741	0.746	0.740	0.740
Long-OOD (32k-65k)	0.463	0.741	0.736	0.314	0.741	0.741	0.736	0.699	0.741	0.741	0.741	0.741	0.745	0.732	0.750
Full (0k-65k)	0.741	0.744	0.741	0.394	0.744	0.744	0.743	0.700	0.739	0.744	0.744	0.744	0.745	0.743	0.739

C.2 Ablation Study for Synthetic Training Data Generation

As mentioned in the description of the GR dataset, the base version of the GR dataset is too small to finetune the ARMT model from scratch. In this section, we provide additional results with the base GR dataset without additional synthetic data; the results are provided in Table 12. One can notice

Table 12: Best results on the GR dataset for Gemma-3-1B-IT model, metric - ROUGE-L.

Model/ Lengths	**No Fine-, Tuning**	**Base, GR, 8k**	**ARMT, GR, 2k**	**ARMT, GR, 4k**	**ARMT, GR, 8k**
0k-1k	0.188	0.347	0.340	0.326	0.313
1k-2k	0.161	0.358	0.175	0.198	0.211
2k-4k	0.140	0.305	0.151	0.184	0.200
4k-6k	0.114	0.288	0.135	0.175	0.180
6k-8k	0.113	0.289	0.146	0.164	0.176
8k-10k	0.106	0.271	0.146	0.152	0.164
10k-12k	0.101	0.232	0.132	0.166	0.177
12k-14k	0.091	0.241	0.120	0.147	0.156
14k-16k	0.088	0.218	0.128	0.149	0.168
16k-24k	0.083	0.222	0.128	0.145	0.164
24k-32k	0.078	0.149	0.111	0.138	0.175
32k-49k	0.078	0.130	0.123	0.135	0.144
49k-65k	0.103	0.072	0.178	0.190	0.201
In-Domain (0k-8k)	0.143	0.317	0.189	0.209	0.216
OOD (8k-65k)	0.092	0.226	0.130	0.150	0.165
Long-OOD (32k-65k)	0.084	0.117	0.136	0.148	0.157
Full (0k-65k)	0.116	0.269	0.157	0.178	0.189

that the base version of the GR dataset contains enough samples to finetune the Gemma-3-1B-IT model, but not enough to finetune the ARMT model.

C.3 Ablation Study for Continuous Pretraining

We also conducted additional pretraining experiments for the ARMT model. We hypothesize that training with the cold-start weights for associative blocks and memory tokens could limit the performance of the model, which is trained on limited data for fine-tuning. To check this, we created an additional synthetic QA dataset with LLM's generated QA pairs over natural long-context samples. We continuously pretrained the ARMT model with the Gemma-3-1B-IT backbone and finetuned it on the GR-100+ dataset with the standard curriculum learning setup. Due to limited resources, we pretrained the model only on the two segments of 1024 tokens. The results are presented in Table 13.

While this training setup allows us to achieve higher performance on in-domain lengths compared to the non-pretrained ARMT, the performance on long-context samples is worse. We hypothesize that this performance drop could arise from the limited length of texts in pretraining. Unfortunately, we cannot conduct pretraining on longer texts due to limited resources. However, this training setup shows promising results and is left for future research.

Table 13: Best results on the GovReport-100+ dataset for Gemma-3-1B-IT model, comparison with the pretraining on synthetic QA task, metric - ROUGE-L.

Model/ Lengths	**No Fine-, Tuning**	**Base, GR-100+, 8k**	**ARMT, GR-100+, 8k**	**ARMT, GR-100+, 8k, Pretrained**
0k-1k	0.188	0.380	0.358	0.363
1k-2k	0.161	0.385	0.303	0.332
2k-4k	0.140	0.352	0.279	0.282
4k-6k	0.114	0.344	0.303	0.284
6k-8k	0.113	0.296	0.254	0.278
8k-10k	0.106	0.306	0.235	0.244
10k-12k	0.101	0.269	0.241	0.251
12k-14k	0.091	0.280	0.237	0.242
14k-16k	0.088	0.231	0.240	0.252
16k-24k	0.083	0.269	0.252	0.249
24k-32k	0.078	0.247	0.207	0.214
32k-49k	0.078	0.171	0.200	0.183
49k-65k	0.103	0.189	0.264	0.018
In-Domain (0k-8k)	0.143	0.351	0.299	0.307
OOD (8k-65k)	0.092	0.266	0.238	0.241
Long-OOD (32k-65k)	0.084	0.175	0.215	0.145
Full (0k-65k)	0.116	0.306	0.266	0.272

D TRAINING HYPERPARAMETERS

The hyperparameters used during training are described in Table 14. All models were trained using two NVIDIA H100-80GB GPUs. The full ARMT fine-tuning process takes approximately 48 hours per run.

Table 14: Training hyperparameters. All models were trained with LoRA (Hu et al., 2021) on all linear layers with rank=64, α=128 and dropout=0.1. We used gradient clipping to the maximum value of 1.0 during training, and weight decay of 0.01. For ARMT model, we used 16 memory tokens and associative memory size of 64.

Dataset	**Model**	**Length**	**Learning Rate**	**Training Steps**	**Total Batch Size**
MT	Gemma-3-1B-IT	8k	1e-4	10.000	64
MT	ARMT	2k	1e-4	10.000	64
MT	ARMT	4k	1e-4	10.000	64
MT	ARMT	8k	3e-5	10.000	64
GR	Gemma-3-1B-IT	8k	1e-4	8.000	64
GR	ARMT	2k	1e-4	4.000	64
GR	ARMT	4k	3e-5	4.000	64
GR	ARMT	8k	1e-5	4.000	64
GR-100+	Gemma-3-1B-IT	8k	1e-4	10.000	64
GR-100+	ARMT	2k	1e-4	5.000	64
GR-100+	ARMT	4k	3e-5	5.000	64
GR-100+	ARMT	8k	1e-5	5.000	64