

Mem-LM: An LM-style Model for Reinforced Memory Selection

Yanhan Huang¹ Yihao Zhong² Wenwen Ouyang³ Xing Tian^{4*}
Zining Yin⁵ Yiyi Lu⁶

¹ Portland Institute, Nanjing University of Posts and Telecommunications, Jiangsu, China

² New York University, NY, United States

³ Independent Researcher, TX, United States

⁴ Independent Researcher, Shanghai, China

⁵ Google, CA, United States

⁶ Duke University, NC, United States

Abstract

External memory banks can improve test-time reasoning in frozen large language models without updating backbone parameters. Yet similarity-based retrievers and pointwise selectors score memories independently, ignoring interactions and presentation order. We introduce MEM-LM, an LM-style model for reinforced memory selection that casts context construction as autoregressive index decoding. From a densely retrieved candidate pool, a lightweight Transformer encoder-decoder generates an ordered index sequence $[i_1, \dots, i_k]$, modeling dependencies among memories while leaving the backbone frozen. This formulation supports standard language-model training. For sparse binary verifiable rewards, we further propose MEM-PO, a confidence-aware extension of group-relative policy optimization that promotes correct, low-confidence rollouts, penalizes confidently incorrect ones, and retains learning signals in reward-homogeneous groups. Across three mathematical reasoning and two reasoning-intensive question-answering benchmarks with three Qwen backbones, MEM-LM achieves an average score of 84.9 on Qwen-plus, exceeding the strongest memory-augmented baseline by 2.3 points.

1 Introduction

External memory can improve the test-time reasoning of frozen large language models (LLMs) without updating their backbone parameters. Such memory banks store past cases, external knowledge, or meta-experience that can be retrieved as in-context demonstrations for a new query (Zhong et al., 2024; Zhou et al., 2025). Consequently, per-

formance depends not only on the backbone but also on the quality of the constructed context.

Existing methods, however, provide limited control over memory composition. Similarity-based retrievers and pointwise selectors evaluate candidates independently, overlooking interactions among selected items and their presentation order (Xu et al., 2026). Both factors can substantially affect a frozen model. Moreover, training a selector with sparse binary rewards is difficult because such rewards poorly distinguish competing selections and yield no relative signal when all rollouts in a group receive the same reward.

We introduce MEM-LM, an LM-style model for reinforced memory selection that formulates context construction as *autoregressive index decoding*. As illustrated in Figure 1, MEM-LM first retrieves a candidate pool and then uses a lightweight Transformer encoder-decoder to generate an ordered index sequence $[i_1, \dots, i_k]$. This formulation captures dependencies among memories while keeping the backbone frozen and supports standard language-model training recipes. We further propose MEM-PO, a confidence-aware extension of group-relative policy optimization. MEM-PO favors correct but low-confidence rollouts, penalizes confidently incorrect ones, and preserves informative gradients in reward-homogeneous groups.

We evaluate MEM-LM on three mathematical reasoning benchmarks (Math-500, AIME-24, and AIME-25) and two reasoning-intensive QA benchmarks (MMLU-Pro and GPQA) using three Qwen backbones. On Qwen-plus, MEM-LM achieves an average score of 84.9, outperforming the strongest memory-augmented baseline by 2.3 points. Our contributions are as follows:

- We formulate memory selection as autoregressive index decoding, enabling MEM-LM to model both the composition and order of memories for a frozen backbone.

* Corresponding author. For any inquiries, please contact: albert_knows@163.com (Xing Tian), p23000626@njupt.edu.cn (Yanhan Huang), yz7654@nyu.edu (Yihao Zhong), wenweno@alummi.cmu.edu (Wenwen Ouyang), zy2605@caa.columbia.edu (Zining Yin), y1996@duke.edu (Yiyi Lu).

- We introduce MEM-PO, a confidence-aware policy optimization method that improves learning from sparse verifiable rewards, including reward-homogeneous rollout groups.
- Experiments on five benchmarks and three Qwen backbones show consistent improvements, while ablations identify ordered decoding and MEM-PO as the main contributors.

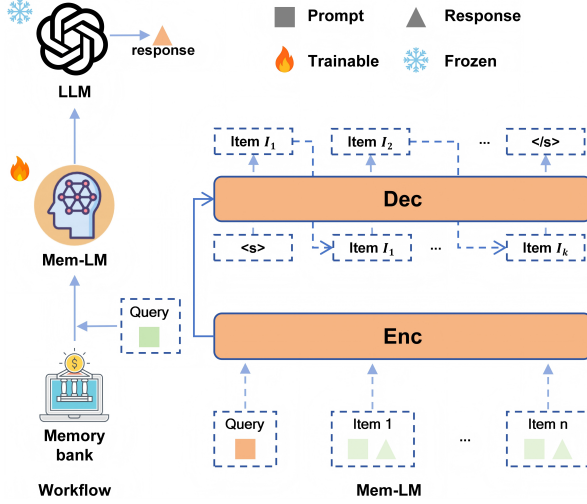


Figure 1: Schematic illustration of our Mem-LM framework.

2 Related Work

Context Optimization for Test-Time Scaling. Recent work augments LLM and VLM agents with explicit memory to support long-horizon reasoning and continual adaptation. Beyond short-term prompting schemes such as ReAct-style interaction and reflection, memory-augmented systems store past traces, plans, or intermediate results for later reuse (Zhong et al., 2024). Other studies introduce structured or semantic memory with explicit add, update, and delete operations (Xu et al., 2026; Salama et al., 2025). A related line distills experience into rules, guidelines, or reusable workflow fragments (Zhao et al., 2024; Fu et al., 2024; Wang et al., 2024b). However, many methods still depend on static similarity-based retrieval or unfiltered memory accumulation, which can hurt scalability and relevance. Memento (Zhou et al., 2025) formulates memory use as a memory-augmented MDP and learns a utility-aware selection policy without tuning the backbone model. Similarly, Qin et al. (2024) learn a query-conditioned selector for case retrieval.

Agentic memory selection is also related to in-context learning, where the key problem is to construct an effective demonstration sequence. Standard approaches retrieve demonstrations with pre-trained embeddings and vector search (Li et al., 2024), or train retrievers with classification (Li et al., 2023) or response-similarity objectives (Hu et al., 2022), extending earlier example-selection methods such as EPR (Rubin et al., 2022). Beyond retrieval, Lever-LM (Yang et al., 2024b) uses a small language model to decode demonstration sequences, VICL (Zhou et al., 2024) adds LLM-based reranking, and SURf (Sun et al., 2024) refines demonstrations through self-feedback. Qin et al. (2024) further compose diverse sequences for each test input. These studies all optimize the test-time context, but they usually under-model the ordered and combinatorial dependencies among selected items. Mem-LM addresses this limitation by autoregressively decoding memory indices.

Reinforcement Learning. Large reasoning models such as OpenAI o1 (Jaech et al., 2024) and DeepSeek-R1 (Guo et al., 2025) show that reinforcement learning can promote self-verification and iterative refinement, enabling inference-time scaling. GRPO (Shao et al., 2024) is a representative method that removes the value function and estimates advantages within response groups. Since the training details of proprietary systems are not fully disclosed, recent open efforts aim to reproduce and improve these recipes. OpenReasoner-Zero (Hu et al., 2026) and Reinforce++ (Hu et al., 2025) provide open training variants, with the latter using global advantage normalization without a critic. VinePPO (Kazemnejad et al., 2024) improves credit assignment for mathematical reasoning, RLEF (Gehring et al., 2024) uses execution feedback for code generation, and DAPO (Yu et al., 2026) extends GRPO to alleviate entropy collapse, sparse rewards, and training instability (Zhang et al., 2025). Built on this line, our proposed Mem-PO adapts group-relative policy optimization to reinforced memory selection in Mem-LM.

3 MemGen Framework

We now introduce our Mem-LM framework in detail.

3.1 Formulation of agentic memory

Given a LLM/VLM agent, denoted as $\mathcal{M}$ (e.g., Deepseek-R1 (Guo et al., 2025)), we would like to adaptively select a list of elements from the agent memory bank, $\mathcal{Z} = \{z_1, \dots, z_n\}$. This sequence $\mathcal{Z}$ consists of n elements (n -shot), where each element z_i is either a previous case, an piece of external knowledge, or a piece of meta-experience (Zhou et al., 2025). For example, for a question answering task, z_i comprises a prompt, and its corresponding response: $z_i = (\text{prompt}_i, \text{response}_i)$. The context sequence $\mathcal{Z}$ is then concatenated with the test sample $x_{\text{test}} = (\text{prompt}_{\text{test}})$ and fed into the LLM agent $\mathcal{M}$. The model subsequently generates a response $\hat{r}_{\text{test}}$:

$$\hat{r}_{\text{test}} = \mathcal{M}(\mathcal{Z}, x_{\text{test}}). \quad (1)$$

Using an embedding model like the BGE (Xiao et al., 2024), and a vector search engine like Faiss (Douze et al., 2025), we can conduct item retrieval from the memory bank efficiently. The test sample x_{test} is transformed to the vector v_{test} , which serves as the input for the vector search. We then retrieve $n > 0$ candidate demos $\{z_j\}_{j=1}^n$ that have the highest similarity scores with x_{test} .

3.2 Mem-LM

Workflow. Mem-LM operates over a memory bank $\mathcal{D} = \{d_1, \dots, d_N\}$, where each item d_i stores a past case, external knowledge, or a meta-experience (Zhou et al., 2025); a case is a pair $d_i = (T_i, R_i)$ of an input T_i and its target response R_i . Given a query q , Mem-LM first retrieves a candidate set $\mathcal{C} = \{d_1, \dots, d_n\} \subseteq \mathcal{D}$ via dense vector search (Faiss (Douze et al., 2025)), and then *composes* an ordered sequence of $k \leq n$ items $[d_{i_1}, \dots, d_{i_k}]$ that is concatenated with q to prompt the frozen LLM/VLM (Fig. 1).

Architecture. Mem-LM couples a frozen embedder $E(\cdot)$ with a trainable Transformer encoder-decoder $M(\cdot)$. The embedder (e.g., BGE (Xiao et al., 2024) or CLIP (Radford et al., 2021)) maps the query and each candidate to fixed-length vectors; for multimodal items the image and text streams are encoded separately and concatenated, giving $h_q = E(I_q, T_q)$ and $h_i = E(I_i, [T_i, R_i])$ for $i = 1, \dots, n$. Candidate embeddings are precomputed at index-building time and reused at inference, incurring negligible overhead.

Index decoding. $M(\cdot)$ takes the concatenated sequence $[h_q; h_1; \dots; h_n]$ as input and, unlike a

GPT-style decoder that emits text tokens, autoregressively decodes a sequence of *memory indices* $[i_1, \dots, i_k]$ with $i_t \in \{1, \dots, n\}$, each pointing to an item in $\mathcal{C}$. By modeling the selection as ordered index generation, Mem-LM captures the sequential and combinatorial dependencies among chosen items that similarity-based retrieval ignores, while leaving the backbone untouched.

3.3 Training Methods for Mem-LM

Because Mem-LM decodes memory indices autoregressively, it can be trained with the standard recipes for language models (Guo et al., 2025; Ouyang et al., 2022): supervised fine-tuning (SFT), reinforcement learning with verifiable rewards (RLVR), and an SFT-then-RLVR pipeline, all formalized in Appendix A. The prevailing RLVR objective, GRPO (Shao et al., 2024), however, transfers poorly to memory selection, which motivates MEMPO, a plug-in modification that augments the group-relative advantage (Eq. 6) with a confidence-aware self-evaluation signal.

Motivation. RLVR rewards are sparse: the format reward (Eq. 12) saturates early, while the accuracy reward is binary and yields little gradient, hindering convergence for small models (Srivastava and Aggarwal, 2025). Model confidence is also poorly calibrated, as models are often overconfident on erroneous reasoning (Sun et al., 2025) yet under-confident on unfamiliar inputs (Kang et al., 2025). We therefore seek a denser signal that *rewards correct but low-confidence rollouts* and *penalizes incorrect but high-confidence ones*.

Mem-PO. Given G rollouts $\{o_i\}_{i=1}^G$ for a query q , let $c_i \in \{0, 1\}$ denote verifiable correctness and $\kappa_i = \frac{1}{L_i} \sum_{t=1}^{L_i} \pi_\theta(o_i[t] \mid o_i[:t-1], q)$ the mean token confidence over the L_i tokens of o_i . We organize the group as a *confidence tournament*: for an ordered pair (o_i, o_j) the preference target is

$$s_{ij} = \begin{cases} +1, & c_i > c_j, \\ \text{sign}(\kappa_j - \kappa_i), & c_i = c_j, \\ -1, & c_i < c_j, \end{cases} \quad (2)$$

so that correctness dominates while, *among equally correct traces, the less-confident one is preferred*. Aggregating pairwise outcomes yields a Bradley-Terry-style self-evaluation score

$$\text{SES}_i = \frac{1}{G-1} \sum_{j \neq i} s_{ij} \sigma\left(\frac{\kappa_i - \kappa_j}{\tau}\right), \quad (3)$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ and $\tau > 0$ is a temperature. We standardize the score and add it to the GRPO advantage,

$$\begin{aligned}\widetilde{\text{SES}}_i &= \frac{\text{SES}_i - \text{mean}(\{\text{SES}_j\}_{j=1}^G)}{\text{std}(\{\text{SES}_j\}_{j=1}^G)}, \\ \tilde{A}_{i,t} &= \hat{A}_{i,t} + \lambda \widetilde{\text{SES}}_i\end{aligned}\quad (4)$$

with $\lambda > 0$; all remaining components follow DAPO (Yu et al., 2026).

Discussion. MemPO turns the binary reward into a denser, confidence-aware signal: it raises the advantage of correct yet low-confidence rollouts above their over-confident peers and suppresses confidently wrong reasoning, producing more discriminative gradients. Crucially, when a group is reward-homogeneous ($\bar{c} \in \{0, 1\}$) the vanilla advantage $\hat{A}_{i,t}$ collapses to zero (Zhou et al., 2026), whereas the correctness-tie branch keeps $\widetilde{\text{SES}}_i$ non-degenerate, recycling otherwise-wasted samples into informative gradients and improving RLVR convergence for small models.

4 Experiments

4.1 Datasets and memory construction

Math problem solving tasks Three math problem solving tasks: Math-500 (Lightman et al., 2024), AIME-24¹, and AIME-25².

Question-answering tasks requiring reasoning

Two challenging question-answering tasks in the science domain: MMLU-pro (Wang et al., 2024a), GPQA (Rein et al., 2023).

We use the Mixture-of-thoughts³ datasets. It consists of a large scale set of (problem, reasoning trace, answer) tuples generated by the Deepseek-R1 model. We only employ the following subset as the memory bank: 5000 samples for Math, and 5000 for other science domains including physics, chemistry, biology, etc. We also draw another 5000 samples for the optimization of our Mem-LM framework.

4.2 Baselines

We compare our framework with the following baselines: (a) IO, that is, directly outputting the final answer. (b) CoT (Wei et al., 2022). These two methods do not utilize the memory bank. (c) ComplexCoT (Fu et al., 2022). (d) OPRO (Yang

et al., 2024a). (e) sparse retrieval using BM25 (Chen et al., 2017) (denoted as Sparse-Retrieval). (f) Dense-Retrieval (Karpukhin et al., 2020). (g) MOMENTO (Zhou et al., 2025). (h) UDR (Li et al., 2023). (i) Lever-LM (Yang et al., 2024b).

4.3 Experimental setups

Devices. All experiments are run on a single machine equipped with NVIDIA A800 GPUs with 80GB memory.

LLM backbones. For language-based tasks, we evaluate three backbone LLMs: Qwen3-8B, Qwen3-32B (Yang et al., 2025), and Qwen-plus⁴, which is accessed through the Alibaba Cloud API. On MMLU-Pro and GPQA, the backbone uses greedy decoding with temperature 0, and a prediction is correct if the extracted option matches the reference answer. For the Math-500, AIME-24, and AIME-25 tasks, we report empirical pass@ k . For each problem, we independently sample k responses using temperature 1.0 and top- $p = 0.95$. A problem is counted as solved if at least one of the k responses yields the correct final answer. We use pass@4 for Math-500 and pass@16 for AIME-24 and AIME-25.

Settings for MEM-LM. For language-based tasks, the embedder $E(\cdot)$ is initialized with BGE-base-en (Xiao et al., 2024). We attach a trainable module $M(\cdot)$ consisting of a one-layer Transformer encoder and a one-layer Transformer decoder. The hidden size of $M(\cdot)$ is set to match the output dimension of the embedder.

Settings for memory selection. We first use Dense-Retrieval to retrieve $n = 64$ candidate memory items. Then, MEM-LM autoregressively decodes the final ordered sequence of $k = 4$ memory items.

Training setups. We implement the training pipeline based on OPEN-R1⁵ for efficient RLVR training. MEM-LM is trained with supervised fine-tuning followed by our MEM-PO optimization. Detailed hyperparameters are reported in Appendix B.1.

4.4 Main Results

Table 1 reports the Qwen-plus results. MEM-LM achieves the best performance on all five benchmarks, with an average score of 84.9. It outperforms Memento, the strongest baseline on aver-

¹<https://huggingface.co/datasets/math-ai/aime24>

²<https://huggingface.co/datasets/math-ai/aime25>

³<https://huggingface.co/datasets/open-r1/Mixture-of-Thoughts>

⁴<https://help.aliyun.com/zh/model-studio/qwen-api-reference/>

⁵<https://github.com/huggingface/open-r1>

Method	Math-500 pass@4	AIME-24 pass@16	AIME-25 pass@16	MMLU-Pro acc	GPQA acc	Avg.
IO	87.5	0.0	0.0	61.6	69.3	43.7
COT	92.5	76.3	70.0	74.4	72.5	77.1
ComplexCoT	91.9	68.2	65.9	75.1	72.6	74.7
Sparse-Retrieval	92.9	79.0	78.9	79.5	75.2	81.1
Dense-Retrieval	93.9	82.0	74.2	80.4	75.2	81.1
Reranker	93.8	78.6	70.8	80.1	74.3	79.5
UDR	93.7	71.4	69.7	80.2	76.4	78.3
Lever-LM	93.8	78.5	75.9	80.3	76.8	81.3
MOMENTO	94.7	83.0	77.8	81.1	76.6	82.6
MEM-LM	96.6	85.5	81.3	82.9	78.4	84.9

Table 1: Overall performance on the math solving and QA benchmarks, on the Qwen-plus LLM backbone.

age, by 2.3 points and the memory-free CoT baseline by 7.8 points. Relative to Memento, the gains are +1.9, +2.5, +3.5, +1.8, and +1.8 points on Math-500, AIME-24, AIME-25, MMLU-Pro, and GPQA, respectively. The largest improvement occurs on AIME-25, indicating the benefit of learned memory composition on a challenging mathematical reasoning task.

The results also generalize across model scales. As shown in Table 9 in Appendix B, MEM-LM consistently outperforms CoT, Dense-Retrieval, and Lever-LM with both Qwen3-8B-Instruct and Qwen3-32B-Instruct, achieving the highest score on every evaluated dataset and in the overall average. These results suggest that similarity-based retrieval alone cannot fully exploit the available memory bank. MEM-LM instead learns both which items to select and how to order them for the frozen backbone. The following ablations separately evaluate the contributions of ordered index decoding and the confidence-aware MEM-PO training objective.

Variant	MMLU-Pro acc	Math-500 pass@4
Mem-LM (Full)	82.9	96.6
w/o ordering (set selection)	81.4	95.1
w/o encoder	81.8	95.4
w/o SFT init.	82.6	96.3
Dense-Retrieval (no selector)	80.4	93.9

Table 2: Ablation on the MEM-LM framework (Qwen-plus backbone). Each row removes one component from the full model.

4.5 Ablation studies and further analysis

Latency analysis. We evaluate the online per-query latency of **Mem-LM** under the same hardware and inference configuration used for Dense-Retrieval. Both methods perform the same query-

n (cand.)	k (sel.)	MMLU-Pro	Math-500
16	4	81.6	95.3
32	4	82.3	96.0
64	4	82.9	96.6
128	4	82.7	96.4
64	2	81.7	95.5
64	4	82.9	96.6
64	6	82.5	96.3
64	8	82.0	95.8

Table 3: Effect of the candidate pool size n and the number of selected memory items k on MEM-LM (Qwen-plus backbone).

Component	Dense-Retrieval	Mem-LM
Query encoding	42.7	42.7
Faiss vector search ($n = 64$)	11.5	11.5
Autoregressive decoding ($k = 4$)	–	8.7
Total online latency	54.2	62.9

Table 4: Per-query online latency of Dense-Retrieval and **Mem-LM**. All values are reported in milliseconds (ms). Candidate embeddings are precomputed offline and are therefore excluded from both methods.

encoding and candidate-retrieval operations: the query is encoded by BGE-base-en, and the top $n = 64$ candidates are retrieved through Faiss vector search. After these shared operations, **Mem-LM** introduces one additional stage that autoregressively decodes an ordered sequence of $k = 4$ memory indices. Table 4 reports the latency breakdown for these online components.

Relative to Dense-Retrieval, **Mem-LM** increases the online retrieval-and-selection latency by only 8.7 ms, from 54.2 to 62.9 ms, corresponding to a relative overhead of $8.7/54.2 = 16.1\%$. This increase is entirely attributable to the autoregressive index-decoding stage, since query encoding and Faiss search are identical for the two methods. Moreover, downstream reasoning and answer generation by the frozen LLM typically require more

than 10 seconds per query in our setting. The additional 8.7 ms therefore amounts to less than 0.1% of the downstream generation time, indicating that **Mem-LM** achieves its performance improvements with a modest practical latency cost.

Ablation on the Mem-LM framework Table 2 isolates each design choice. Replacing ordered autoregressive decoding with unordered *set selection* causes the sharpest drop (−1.5 on MMLU-Pro, −1.5 on Math-500), confirming that the *order* of composed items is the dominant source of our gains, precisely the dependency pointwise retrieval cannot capture. Removing the Transformer *encoder*, so the decoder reads raw candidate embeddings, eliminates cross-candidate interaction and yields a smaller but consistent loss (−1.1 on MMLU-Pro, −1.2 on Math-500), indicating that jointly encoding the query and the pool helps the decoder reason about item complementarity. Omitting the *SFT initialization* and training Mem-PO from scratch costs only −0.3 and −0.3, so SFT mainly speeds convergence and fixes the output format, while the reinforcement objective drives final accuracy. Every variant still beats the Dense-Retrieval baseline, showing that all components help and that ordered, interaction-aware decoding is the most critical.

We next vary the candidate pool size n and the number of selected items k (Table 3). Accuracy rises as n grows from 16 to 64 and then saturates, slightly declining at $n=128$, so a moderate pool gives enough coverage without adding distractors. For k , performance peaks at $k=4$: fewer items ($k=2$) under-use the memory bank, whereas more ($k=8$) inject noise and lengthen the context. We thus set $n=64$ and $k=4$ by default.

Method	MMLU-Pro	Math-500
	acc	pass@4
SFT	80.3	93.7
PPO	80.7	94.5
REINFORCE	79.7	93.9
GRPO	80.9	95.0
DAPO	81.1	95.2
Mem-PO (ours)	82.9	96.6

Table 5: Ablation on the training method for Mem-LM.

Ablation on the training method for Mem-LM Table 5 compares optimizers for Mem-LM: SFT, REINFORCE, PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), DAPO (Yu et al., 2026), and Mem-PO. SFT is weakest (80.3/93.7 on MMLU-Pro/Math-500), as imitating fixed ora-

cle sequences ignores the query-dependent utility of memory combinations. All RLVR variants surpass it, confirming that reward-driven exploration is essential. REINFORCE and PPO remain near SFT, while the group-relative GRPO and DAPO are clearly stronger (80.9/95.0 and 81.1/95.2) thanks to more stable relative advantages under sparse binary rewards. Mem-PO is best on both benchmarks (82.9/96.6), beating DAPO by +1.8 and +1.4. The gain comes from its confidence-aware self-evaluation signal, which densifies the binary reward and, via the correctness-tie branch, recovers gradients from reward-homogeneous groups where the group-relative advantage vanishes.

Embedder model	MMLU-Pro
all-mpnet-base-v2	82.2
Contriever	81.9
GTE-base	82.5
GTE-large	83.0
BGE-base-en	82.9
BGE-large-en	83.2

Table 6: Ablation on the embedder model $E(\cdot)$ (Qwen-plus backbone, MMLU-Pro).

Confidence-direction ablation. We test whether the gains of MEM-PO arise merely from adding confidence or from preferring the appropriate confidence direction. The full objective favors lower-confidence rollouts within correctness ties, thereby reinforcing uncertain successes while suppressing confidently incorrect selections. The *high-confidence preference* variant reverses only the correct-tie branch:

$$s_{ij}^{\text{high}} = \begin{cases} \text{sign}(\kappa_i - \kappa_j), & c_i = c_j = 1, \\ s_{ij}, & \text{otherwise,} \end{cases} \quad (5)$$

whereas the *no-confidence* variant sets $\lambda = 0$ in $\tilde{A}_{i,t} = \hat{A}_{i,t} + \lambda \text{SES}_i$ and thus recovers the DAPO-style objective. GRPO is included as an additional reference. All variants use the same SFT initialization, training data, rollout budget, optimization settings, and evaluation protocol.

The low-confidence preference achieves 82.9 accuracy on MMLU-Pro and 96.6 pass@4 on Math-500, exceeding the high-confidence alternative by 1.5 and 0.9 points, respectively. Because these variants differ only in the correct-tie branch, the result shows that reinforcing correct but under-supported memory sequences is more effective than amplifying successes already favored by the selector. Removing the confidence term causes larger drops of

Method	MMLU-Pro	Math-500
	acc	pass@4
MEM-PO (low-conf. preference)	82.9	96.6
High-confidence preference	81.4	95.7
No confidence signal ($\lambda = 0$)	81.1	95.2
GRPO reference	80.9	95.0

Table 7: Ablation of the confidence-preference direction in MEM-PO with the Qwen-plus backbone. Results are averaged over five random seeds.

Method	Accuracy (%)
CoT	74.1
Dense-Retrieval	89.2
UDR	90.7
Lever-LM	91.8
MOMENTO	90.5
Mem-LM	93.3

Table 8: Diagnosis accuracy on the internal MED-DIAGNOSIS dataset using Qwen-plus as the frozen backbone.

1.8 and 1.4 points, confirming that the auxiliary signal itself is also beneficial. Together, these results support both components of MEM-PO: confidence-aware advantage shaping and its specific emphasis on uncertain successes.

Ablation on the embedder $E(\cdot)$ We test Mem-LM’s sensitivity to the frozen embedder on MMLU-Pro, fixing the trainable module $M(\cdot)$ and the Mem-PO objective. Swapping in six off-the-shelf encoders—all-mpnet-base-v2 (Reimers and Gurevych, 2019), Contriever (Izacard et al., 2021), GTE-base and GTE-large⁶, and BGE-base-en/BGE-large-en (Xiao et al., 2024)—and re-precomputing only the candidate embeddings, accuracy stays within 81.9–83.2 and every variant exceeds Dense-Retrieval (80.4). Stronger encoders add only marginal gains (BGE-large-en, 83.2), showing that representation quality is not the main driver. Since our default BGE-base-en (82.9) nearly matches the best at lower cost, we adopt it throughout.

Real-world validation on an industrial dataset. We further evaluate MEM-LM on MED-DIAGNOSIS, an internal retrospective dataset developed with nine hospitals. Its 18,719 de-identified cases contain medical histories, examination findings, and reported symptoms. The memory bank combines historical cases with medical knowledge entries, thus requiring selection

across heterogeneous evidence types. All methods use Qwen-plus as the same frozen backbone, and performance is measured by diagnostic-label accuracy against the reference diagnoses.

As shown in Table 8, MEM-LM achieves 93.3% accuracy, outperforming Lever-LM, the strongest baseline, by 1.5 percentage points and exceeding UDR and Dense-Retrieval by 2.6 and 4.1 points, respectively. These results indicate that autoregressively composing an ordered sequence of complementary memories transfers from academic reasoning benchmarks to retrospective clinical data. This experiment evaluates diagnostic-label prediction only; it demonstrates cross-domain transfer but does not establish prospective clinical utility or deployment readiness.

5 Conclusion

This work introduced MEM-LM, an LM-style model for reinforced memory selection that formulates context construction as autoregressive decoding over memory indices. Rather than directly providing a frozen LLM/VLM with an unordered, similarity-ranked set, MEM-LM jointly determines which memories to select and the order in which they should appear, without updating the backbone parameters. We further developed MEM-PO, a modular confidence-aware extension to group-relative policy optimization that provides discriminative training signals under sparse verifiable rewards, including for reward-homogeneous rollout groups. Experiments on five mathematical and reasoning-intensive QA benchmarks with three Qwen backbones demonstrate consistent improvements over retrieval-based and learned memory selectors. On Qwen-plus, MEM-LM achieves an average score of 84.9, exceeding the strongest memory-augmented baseline by 2.3 points; the ablations identify ordered index decoding and the MEM-PO objective as the principal contributors to these improvements. Overall, the results establish LM-style index generation as an effective and modular approach to memory composition, while evaluation on multimodal and longer-horizon agentic tasks remains an important direction for future work.

Limitations

Evaluation scope and external validity. Our evaluation primarily covers five public, text-only benchmarks for mathematical reasoning and

⁶<https://huggingface.co/thenlper/gte-large>

reasoning-intensive question answering, using three backbones from the Qwen family. We additionally report a retrospective evaluation on the internal MED-DIAGNOSIS dataset, but this experiment does not establish prospective clinical utility, deployment readiness, or reproducibility on a publicly accessible medical benchmark. Although the MEM-LM formulation can in principle accommodate multimodal memory items and multi-step agent interactions, we do not evaluate vision-language inputs, interactive environments, or substantially longer reasoning horizons. Consequently, the effectiveness of ordered index decoding in these settings remains to be established.

Optimization cost and hyperparameter sensitivity. MEM-PO uses token-level confidence estimates and performs pairwise comparisons among the G rollouts in each group, resulting in $O(G^2)$ pairwise operations in addition to the underlying group-relative policy optimization. The objective also introduces the weighting coefficient λ and temperature τ , while the broader MEM-LM pipeline requires choices for the candidate-pool size n and the number of selected memories k . Although our ablations indicate reasonable robustness around the evaluated configuration, these settings may require validation or re-tuning when the memory distribution, task domain, or backbone changes. Moreover, our latency measurements use a single hardware configuration with $n = 64$ and $k = 4$; the observed overhead may therefore not generalize to substantially larger candidate pools, selected contexts, or rollout groups.

Dependence on confidence quality and memory coverage. The auxiliary signal in MEM-PO is derived from the token probabilities of the trainable selection policy π_θ . Because these probabilities are not guaranteed to be calibrated, confidence estimates that are systematically misaligned with selection quality could reduce the benefit of the proposed advantage shaping. In addition, most memory items used in our benchmark experiments are constructed from DeepSeek-R1-generated traces. Their accuracy, diversity, and domain coverage constrain the evidence available to MEM-LM, and the initial dense retriever may exclude useful items before autoregressive selection begins. Performance may consequently degrade for noisy memory banks, low-resource domains, or queries that differ substantially from the memory construction distribution; evaluating more diverse memory sources and retrieval mechanisms is left for future

work.

Declaration for Using AI Assistants

During the preparation of this work, AI-assisted tools were used to support language polishing, grammar correction, and improvement of the clarity and readability of the text. The AI assistants were not used to generate original research ideas, perform data analysis, draw conclusions, or replace the authors’ own critical thinking and academic judgment. All AI-generated suggestions were carefully reviewed, revised, and verified by the authors. The authors take full responsibility for the accuracy, integrity, and originality of the content presented in this work.

References

- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.
- Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The faiss library. *IEEE Transactions on Big Data*.
- Yao Fu, Dong-Ki Kim, Jaekyeom Kim, Sungryull Sohn, Lajanugen Logeswaran, Kyunghoon Bae, and Honglak Lee. 2024. Autoguide: Automated generation and selection of context-aware guidelines for large language model agents. *Advances in Neural Information Processing Systems*, 37:119919–119948.
- Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.
- Jonas Gehring, Kunhao Zheng, Jade Copet, Vegard Mella, Quentin Carbonneaux, Taco Cohen, and Gabriel Synnaeve. 2024. Rlef: Grounding code llms in execution feedback with reinforcing learning. *arXiv preprint arXiv:2410.02089*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Jian Hu and 1 others. 2025. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 1(4).
- Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2026. Open-reasoner-zero: An open source approach to scaling

- up reinforcement learning on the base model. *Advances in Neural Information Processing Systems*, 38:162239–162262.
- Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A Smith, and Mari Ostendorf. 2022. In-context learning for few-shot dialogue state tracking. In *Findings of the association for computational linguistics: EMNLP 2022*, pages 2627–2643.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.
- Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2025. Unfamiliar finetuning examples control how language models hallucinate. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3600–3612.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.
- Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordani, Siva Reddy, Aaron Courville, and Nicolas Le Roux. 2024. Vineppo: Refining credit assignment in rl training of llms. *arXiv preprint arXiv:2410.01679*.
- Li Li, Jiawei Peng, Huiyi Chen, Chongyang Gao, and Xu Yang. 2024. How to configure good in-context sequence for visual question answering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26700–26710. IEEE.
- Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4644–4668.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let’s verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- Chengwei Qin, Aston Zhang, Chen Chen, Anirudh Dagar, and Wenming Ye. 2024. In-context learning with iterative demonstration selection. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7441–7455.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.
- Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2655–2671.
- Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. 2025. Meminsight: Autonomous memory augmentation for llm agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33124–33140.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Saksham Sahai Srivastava and Vaneet Aggarwal. 2025. A technical survey of reinforcement learning techniques for large language models. *ACM Transactions on Intelligent Systems and Technology*.
- Fengfei Sun, Ningke Li, Kailong Wang, and Lorenz Goette. 2025. Large language models are overconfident and amplify human bias. *arXiv preprint arXiv:2505.02151*.

- Jiashuo Sun, Jihai Zhang, Yucheng Zhou, Zhaochen Su, Xiaoye Qu, and Yu Cheng. 2024. Surf: Teaching large vision-language models to selectively utilize retrieved information. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7611–7629.
- Shangshang Wang, Julian Asilis, Ömer Faruk Akgül, Enes Bilgin, Ollie Liu, and Willie Neiswanger. 2026. Tina: Tiny reasoning models via lora. In *International Conference on Learning Representations*, volume 2026, pages 90106–90140.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, and 1 others. 2024a. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.
- Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2024b. Agent workflow memory. *arXiv preprint arXiv:2409.07429*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muenighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2026. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024a. Large language models as optimizers. In *International Conference on Learning Representations*, volume 2024, pages 12028–12068.
- Xu Yang, Yingzhe Peng, Haoxuan Ma, Shuo Xu, Chi Zhang, Yucheng Han, and Hanwang Zhang. 2024b. Lever lm: configuring in-context sequence to lever large vision language models. *Advances in Neural Information Processing Systems*, 37:100341–100368.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaocong Liu, Lingjun Liu, and 1 others. 2026. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.
- Chong Zhang, Yue Deng, Xiang Lin, Bin Wang, Dianwen Ng, Hai Ye, Xingxuan Li, Yao Xiao, Zhanfeng Mo, Qi Zhang, and 1 others. 2025. 100 days after deepseek-r1: A survey on replication studies and more directions for reasoning language models. *arXiv preprint arXiv:2505.00551*.
- Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642.
- WanJun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 19724–19731.
- Han Zhou, Adam X Yang, Laurence Aitchison, Anna Korhonen, and Albert Q Jiang. 2026. Reasoning arena: Trace tournaments when verifiable rewards fall short. *arXiv preprint arXiv:2606.09380*.
- Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, and 1 others. 2025. Memento: Fine-tuning llm agents without fine-tuning llms. *arXiv preprint arXiv:2508.16153*.
- Yucheng Zhou, Xiang Li, Qianning Wang, and Jianbing Shen. 2024. Visual in-context learning for large vision-language models. In *Findings of the association for computational linguistics: ACL 2024*, pages 15890–15902.

A Appendix: Preliminaries on RLVR

RLVR and GRPO. RLVR has become central to LLM research for its ability to incentivize reasoning (Guo et al., 2025). It adopts Group Relative Policy Optimization (GRPO) (Shao et al., 2024), a variant of PPO (Schulman et al., 2017) that discards the value function and instead estimates advantages within a group of sampled responses. For an input–answer pair (q, a) , the old policy $\pi_{\theta_{\text{old}}}$ samples G responses $\{o_i\}_{i=1}^G$ with rewards $\{R_i\}_{i=1}^G$. The advantage of o_i is normalized within the group and shared across its tokens:

$$\hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^G)}{\text{std}(\{R_j\}_{j=1}^G)}. \quad (6)$$

GRPO maximizes a clipped objective with a KL penalty:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a), \{o_i\} \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{G} \text{Obj} - \text{KL} \right], \quad (7)$$

where the per-group objective and penalty are

$$\text{Obj} = \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min(r_{i,t} \hat{A}_{i,t}, \hat{r}_{i,t} \hat{A}_{i,t}), \quad (8)$$

$$\text{KL} = \beta D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{ref}}). \quad (9)$$

Here the importance ratio and its clipped form are

$$r_{i,t} = \frac{\pi_{\theta}(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \quad \hat{r}_{i,t} = \text{clip}(r_{i,t}, 1-\epsilon, 1+\epsilon) \quad (10)$$

Outcome reward. RLVR uses the verifiable task score as the outcome reward,

$$R(\hat{y}, y) = \text{Score}(\hat{y}, y), \quad (11)$$

where y is the ground-truth answer and $\hat{y}$ the prediction. This signal has proven effective at eliciting reasoning across domains such as theorem proving, programming, and mathematics competitions (Guo et al., 2025). For our QA and math tasks, $\text{Score}(\cdot, \cdot)$ is the per-sample accuracy.

Format reward. The reward typically combines correctness with auxiliary constraints. A binary format reward enforces structured outputs:

$$R_{\text{format}}(\hat{y}) = \text{CorrectFormat}(\hat{y}), \quad (12)$$

where $\text{CorrectFormat}(\cdot)$ returns 1 iff the required format holds. For MEM-LM, this requires decoding a valid, non-repeating sequence of memory indices. Following DAPO (Yu et al., 2026), our MEM-PO drops the KL term and adopts asymmetric clipping; Eq. 6 is the group-relative advantage that MEM-PO augments in the main text.

B Appendix: Experiments

B.1 Experimental Settings

We train MEM-LM with an SFT-then-RLVR pipeline, where the RLVR stage uses our MEM-PO objective. From the training set built in Section 4.1, we hold out 300 samples for SFT and 300 for validation, and use the remainder for RLVR. For each SFT sample, the target memory sequence is obtained with the beam-search construction of Yang et al. (2024b). SFT runs for 5 epochs with learning rate 1×10^{-4} and batch size 8, teaching MEM-LM the output format and a reasonable initial selection policy before reinforcement learning.

The RLVR stage follows established practice, in particular OPEN-R1 and TINA (Wang et al., 2026), with minor adjustments for our setting. We

set the weight of $\widetilde{\text{SES}}_i$ to $\lambda=0.5$ and the temperature to $\tau=0.1$. The policy learning rate is fixed to 5×10^{-5} , and the KL coefficient to 0.1 to balance exploration and stability. Per prompt we sample $G=8$ responses with temperature 1.0 and top- $p = 0.95$, using a batch size of 32 per GPU. For policy updates we apply asymmetric clipping with $\epsilon_{\text{high}}=0.28$ and $\epsilon_{\text{low}}=0.20$, and train for 2,000 steps. Unlike DAPO (Yu et al., 2026), MEM-PO retains reward-homogeneous groups, including groups in which all rollouts are correct or all are incorrect. Although the group-relative reward advantage vanishes for such groups, the confidence-based correctness-tie branch can yield a non-degenerate self-evaluation score $\widetilde{\text{SES}}_i$, allowing these samples to contribute informative policy gradients. Checkpoints are saved every 200 steps and evaluated on the validation set; the best-performing checkpoint is used for test-time prediction.

All results are averaged over five random seeds.

B.2 Experiments on More LLM Backbones

Table 9 reports MEM-LM on the three math benchmarks (Math-500, AIME-24, AIME-25) and the two science QA tasks (MMLU-Pro, GPQA) with the Qwen3-8B-Instruct and Qwen3-32B-Instruct backbones. MEM-LM again outperforms both the memory-free CoT baseline and the strongest learned selector (Lever-LM) on every dataset and on average, confirming that its gains transfer across backbone scales.

Method	Math-500 pass@4	AIME-24 pass@16	AIME-25 pass@16	MMLU-Pro acc	GPQA acc	Avg.
<i>Qwen3 8B Instruct</i>						
COT	74.0	0.0	0.0	43.7	42.7	32.1
Dense-Retrieval	77.0	38.2	34.4	43.8	50.9	48.9
Lever-LM	77.1	42.5	36.9	48.6	52.1	51.4
Ours	80.8	45.5	39.9	50.8	56.4	54.7
<i>Qwen3 32B Instruct</i>						
COT	84.6	51.9	43.3	60.0	56.8	59.3
Dense-Retrieval	86.5	55.2	50.5	64.7	62.3	63.8
Lever-LM	86.9	54.6	47.2	66.6	65.0	64.1
Ours	89.6	64.1	54.1	70.2	69.1	69.4

Table 9: Overall performance on the math solving and question-answering benchmarks.