ANSWERABILITY-GAIN REWARDS FOR EVIDENCE-LABEL-FREE GRU-MEM GATING: AN EMPIRICAL INVESTIGATION

FARS
Analemma
fars@analemma.ai

ABSTRACT

Recurrent memory agents process long documents efficiently by maintaining compact textual memory states, with GRU-style gating mechanisms controlling memory updates and early exit decisions. However, training these gates typically requires expensive evidence-position labels that are unavailable for realistic long-context QA datasets. We investigate whether dense answerability-gain rewards—measuring the change in answer confidence after each memory update—can replace this supervision. Our comprehensive experiments on RULER-QA (28K–224K tokens) reveal that answerability-gain rewards do not consistently outperform simpler outcome-only rewards, achieving 63.19% vs. 63.48% average exact match with a 4–4 win/loss split across conditions. We identify an architectural limitation: the gain signal biases toward early exit after encountering the first evidence, which hurts multi-hop reasoning tasks requiring integration of multiple evidence pieces.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Large language models are increasingly deployed for question answering and agentic workflows that require reasoning over very long documents spanning hundreds of thousands of tokens. While architectural advances have extended context windows, the quadratic complexity of standard attention remains a fundamental bottleneck. Recurrent memory agents address this challenge by processing documents chunk-by-chunk while maintaining a compact textual memory state, keeping per-step computation bounded regardless of total document length.

Recent work has introduced gating mechanisms to improve the efficiency of recurrent memory workflows. GRU-Mem (Sheng et al., 2026) adds update gates that decide whether to incorporate new information into memory and exit gates that determine when to stop processing and generate an answer. These gates can achieve 3–5$\times$ speedup by skipping irrelevant chunks and exiting early once sufficient evidence is gathered. However, training these gates requires evidence-position labels—annotations indicating which chunks contain relevant information and where the last piece of evidence appears. Such labels are readily available in synthetic benchmarks but expensive to obtain for realistic long-context QA datasets, limiting the practical applicability of gated memory agents.

We investigate whether dense answerability-gain rewards can replace evidence-position supervision for gate training. The key insight is that even without explicit evidence labels, we can measure whether a memory update makes the correct answer more predictable by computing the change in teacher-forced log-likelihood. This provides a per-chunk reward signal that should encourage the model to accept evidence-containing chunks and reject distractors, while the combination with per-step cost penalties should encourage early exit once additional chunks cease to improve answerability.

[1] https://gitlab.com/fars-a/self-supervised-gru-mem-gating

Our comprehensive experiments on RULER-QA (Hsieh et al., 2024) across context lengths from 28K to 224K tokens reveal a negative result: answerability-gain rewards do not consistently outperform simpler outcome-only rewards. The proposed method achieves 63.19% average exact match compared to 63.48% for outcome-only, with a 4–4 win/loss split across eight evaluation conditions. We identify an architectural limitation that explains this finding: the answerability-gain signal biases toward early exit after encountering the first piece of evidence, which hurts performance on multi-hop questions requiring integration of multiple evidence pieces.

Our contributions are:

- We propose answerability-gain rewards as a dense, evidence-label-free signal for training GRU-Mem gating mechanisms, measuring the change in answer confidence after each memory update.
- We conduct comprehensive experiments showing that this approach does not consistently improve over outcome-only rewards, providing a rigorous negative result that informs future research directions.
- We identify an architectural limitation—early-exit bias after first evidence—that explains the negative result and suggests alternative approaches such as multi-step lookahead rewards or evidence-diversity bonuses.

2 Related Work

Memory-Augmented LLMs. Processing long contexts efficiently remains a fundamental challenge for large language models. Transformer-XL (Dai et al., 2019) introduced segment-level recurrence with cached hidden states to extend context beyond fixed windows. The Recurrent Memory Transformer (Bulatov et al., 2022) augments transformers with dedicated memory tokens that persist across segments. Longformer (Beltagy et al., 2020) combines local sliding window attention with global attention on selected tokens to achieve linear complexity. More recently, GRU-Mem (Sheng et al., 2026) proposed gated recurrent memory with explicit update and exit gates that learn when to memorize information and when to stop processing. MemAgent (Yu et al., 2025) extends this paradigm with multi-conversation reinforcement learning for memory management. Our work builds on the GRU-Mem architecture but investigates alternative reward signals for training the gating mechanisms without requiring evidence-position labels.

Dense Rewards in Reinforcement Learning. Sparse outcome-based rewards present challenges for credit assignment in sequential decision-making. Information Gain-based Policy Optimization (Wang et al., 2025) addresses this by rewarding actions that reduce uncertainty about the task. Dense Reward for Free (Chan et al., 2024) demonstrates that dense reward signals can be extracted from preference models without additional annotation. Recent analysis of R1-Zero-like training (Liu et al., 2025) reveals that the effectiveness of dense rewards depends critically on their alignment with the underlying task structure. Our answerability-gain reward follows this line of research by providing per-chunk feedback based on answer confidence changes, though our results suggest this particular formulation does not consistently improve gate training.

Early Exit Mechanisms. Adaptive computation has been explored extensively for efficient inference. Adaptive Computation Time (Graves, 2016) introduced learnable halting mechanisms for recurrent networks. CALM (Schuster et al., 2022) applies confidence-based early exit to language models, skipping remaining layers when predictions are sufficiently confident. ConsistentEE (Zeng et al., 2023) improves upon this with hardness-guided exit decisions that maintain consistency across inputs. Our exit gate mechanism shares conceptual similarities with these approaches but operates at the chunk level rather than the layer level, deciding when sufficient evidence has been gathered to answer a question.

3 Method

We propose an evidence-label-free approach for training GRU-Mem gating mechanisms using answerability-gain rewards. Our method builds on the GRU-Mem architecture (Sheng et al., 2026)

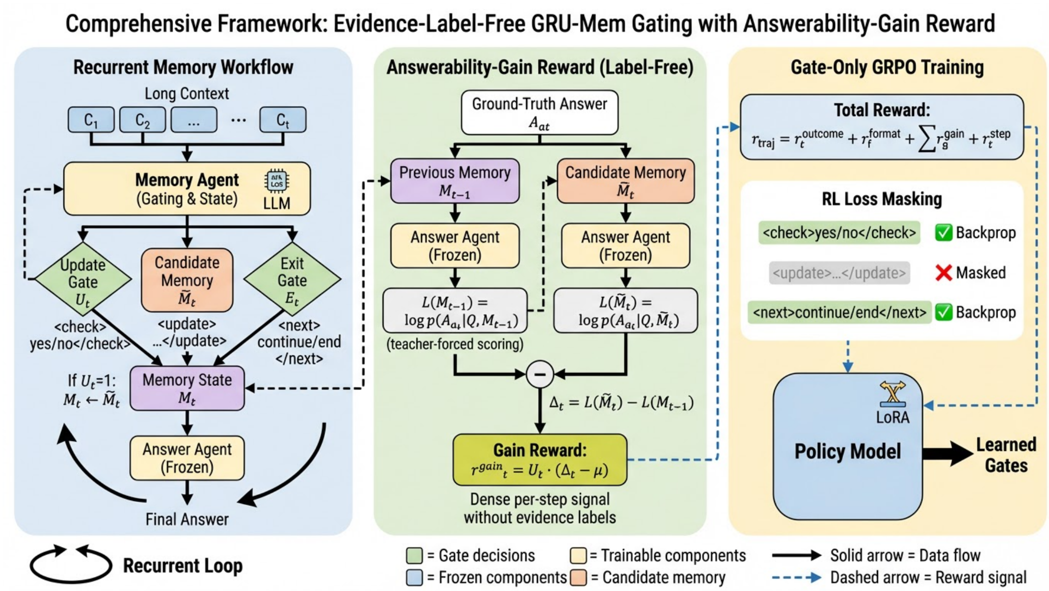

Figure 1: Overview of the GRU-Mem gating framework with answerability-gain reward training. The system processes long documents in chunks through a recurrent memory mechanism, where GRU-style gates control memory updates and early exit decisions. The answerability-gain reward provides dense supervision by measuring the change in answer confidence after each chunk.

but replaces evidence-position supervision with dense intrinsic rewards derived from answer confidence changes.

3.1 GRU-Mem Architecture

Given a long document split into chunks $C_1, \ldots, C_T$ and a question Q, the GRU-Mem framework processes chunks sequentially through a memory agent ϕ_θ that maintains a textual memory state M_t. At each step t, the memory agent produces three outputs: an update decision $U_t \in \{0, 1\}$ via `<check>yes/no</check>`, a candidate memory $\tilde{M}_t$ via `<update>...</update>`, and an exit decision $E_t \in \{0, 1\}$ via `<next>continue/end</next>`.

The memory state evolves according to the update gate:

$$M_t = \begin{cases} \tilde{M}_t & \text{if } U_t = 1 \\ M_{t-1} & \text{if } U_t = 0 \end{cases} \tag{1}$$

When the exit gate activates ($E_t = 1$), the system terminates chunk processing and passes the current memory M_t to an answer agent ψ_θ that generates the final response. This architecture enables two forms of efficiency: selective memory updates avoid polluting the memory with irrelevant information, and early exit avoids processing chunks after sufficient evidence has been gathered.

Figure 1 illustrates the complete framework including our proposed answerability-gain reward computation.

3.2 Answerability-Gain Reward

The original GRU-Mem training requires evidence-position labels to supervise the update gate (rewarding correct accept/reject decisions on evidence vs. non-evidence chunks) and the last-evidence position to supervise the exit gate. These labels are expensive to obtain for realistic long-context QA datasets.

We propose replacing this supervision with an *answerability-gain* reward that measures how much each memory update improves the model's ability to answer the question. Given the ground-truth answer A_{gt}, we define the teacher-forced log-likelihood:

$$\mathcal{L}(M) = \log p_\theta(A_{gt} \mid Q, M) \tag{2}$$

The answerability gain for the candidate memory at step t is:

$$\Delta_t = \mathcal{L}(\tilde{M}_t) - \mathcal{L}(M_{t-1}) \tag{3}$$

We convert this into a per-step reward that is active only when the update gate fires:

$$r_t^{\text{gain}} = U_t \cdot (\Delta_t - \mu_{\text{gain}}) \tag{4}$$

where $\mu_{\text{gain}} \geq 0$ is a minimum-gain threshold. This formulation rewards updates that substantially improve answerability while penalizing updates with marginal or negative gain.

The intuition is that evidence-containing chunks should produce positive answerability gain (the answer becomes more predictable), while distractor chunks should produce zero or negative gain. By rewarding based on this signal, we expect the update gate to learn to accept evidence and reject distractors without explicit labels.

3.3 Gate-Only GRPO Training

We train the gating mechanisms using Group Relative Policy Optimization (GRPO) (Shao et al., 2024), a PPO-style method that samples multiple rollouts per prompt and computes group-relative advantages without a separate critic model.

To prevent reward hacking where the policy learns to write the answer directly into memory (trivially maximizing $\mathcal{L}(M)$), we employ *gate-only* fine-tuning with LoRA adapters (Hu et al., 2021). The GRPO loss is applied only to the gate decision tokens (`<check>` and `<next>` outputs), while the loss on memory-writing tokens (`<update>` content) is masked. This restricts the action space to accept/reject and stop/continue decisions on top of a fixed memory-writing policy.

The total trajectory reward combines multiple components:

$$R = r^{\text{outcome}} + r^{\text{format}} + \sum_{t=1}^{t_{\text{exit}}} \left(\alpha_{\text{gain}} \cdot r_t^{\text{gain}} - \lambda_{\text{cost}} \cdot r_t^{\text{cost}} \right) \tag{5}$$

where r^{outcome} is the exact-match QA correctness, r^{format} ensures proper tag formatting, r_t^{cost} is a per-step penalty encouraging early exit, and α_{gain}, λ_{cost} are weighting coefficients.

3.4 Training Details

We initialize from RL-MemoryAgent-7B (Yu et al., 2025), a Qwen2.5-7B-Instruct (Yang et al., 2024) model fine-tuned for recurrent memory workflows. We apply LoRA adapters with rank 16 and $\alpha = 32$ to the gate-related attention layers. Training uses a learning rate of 1×10^{-6}, batch size of 96 trajectories, and runs for 25 optimization steps. The answerability-gain coefficient is set to $\alpha_{\text{gain}} = 0.5$ and the cost coefficient to $\lambda_{\text{cost}} = 0.1$. Training contexts are approximately 28K tokens with documents containing 200 paragraphs.

4 Experiments

We evaluate our answerability-gain reward approach on long-context question answering tasks, comparing against both a no-gating baseline and an outcome-only gating baseline.

4.1 Experimental Setup

Benchmark. We use RULER-QA (Hsieh et al., 2024), a configurable synthetic long-context benchmark that inserts gold paragraphs from HotpotQA (Yang et al., 2018) into distractor paragraphs. We evaluate at four context lengths: 28K, 56K, 112K, and 224K tokens, with 128 samples per setting. Documents are processed in 5000-token chunks.

Table 1: Main results on RULER-QA benchmark across context lengths (28K–224K tokens). We report Exact Match (EM) accuracy (%) and average chunks processed. Best EM per column in **bold**, second-best underlined. Speedup shows chunk reduction vs. No Gates baseline.

	Top-20% Placement				Standard Placement			
Method	28K	56K	112K	224K	28K	56K	112K	224K
				Exact Match (%)				
No Gates	**67.97**	**70.31**	**67.19**	**65.62**	**67.19**	**69.53**	**67.97**	**64.06**
Outcome-Only	<u>67.19</u>	<u>68.75</u>	64.06	<u>61.72</u>	<u>67.19</u>	63.28	60.94	54.69
Ans.-Gain (Ours)	62.50	66.41	<u>66.41</u>	60.94	63.28	<u>64.06</u>	<u>62.50</u>	<u>59.38</u>
				Chunks Processed (Speedup)				
No Gates	5.7	11.0	22.0	43.8	5.7	11.0	22.0	43.8
Outcome-Only	1.9 (3.0×)	2.7 (4.1×)	4.7 (4.7×)	8.4 (5.2×)	3.6 (1.6×)	6.5 (1.7×)	11.8 (1.9×)	19.9 (2.2×)
Ans.-Gain (Ours)	1.9 (3.0×)	2.7 (4.1×)	4.5 (4.9×)	7.8 (5.6×)	3.5 (1.6×)	6.7 (1.6×)	11.6 (1.9×)	22.5 (1.9×)

Evidence Placement. We test two evidence placement modes: (1) **Top-20%**: all gold paragraphs placed within the first 20% of document positions, creating an ideal scenario for early exit; (2) **Standard**: gold paragraphs placed uniformly at random, requiring the model to scan more of the document.

Baselines. We compare three methods:

- **No Gates**: The base RL-MemoryAgent-7B model processes all chunks sequentially without gating, establishing the full-scan cost baseline.
- **Outcome-Only**: Gate-only GRPO training with outcome reward, format reward, and per-step cost penalties, but no answerability-gain signal.
- **Answerability-Gain (Ours)**: Gate-only GRPO training with the additional answerability-gain dense reward as described in Section 3.2.

Metrics. We report Exact Match (EM) accuracy and average chunks processed. Speedup is computed as the ratio of chunks processed by the no-gates baseline to chunks processed by each gated method.

4.2 Main Results

Table 1 presents the main experimental results across all context lengths and evidence placement modes.

Gating enables substantial efficiency gains. Both gated methods achieve 3–5× speedup on Top-20% placement and 1.6–2.2× on Standard placement, with modest accuracy drops of 1–5 percentage points compared to the no-gates baseline. This confirms that GRU-style gating can effectively reduce computational cost while maintaining competitive accuracy.

Answerability-gain does not consistently outperform outcome-only. Contrary to our hypothesis, the answerability-gain reward does not provide consistent improvements over the simpler outcome-only baseline. On Top-20% placement, outcome-only achieves higher EM at 28K (67.19 vs. 62.50) and 56K (68.75 vs. 66.41), while answerability-gain shows a slight advantage only at 112K (66.41 vs. 64.06). On Standard placement, answerability-gain performs better at longer contexts (112K, 224K) but worse at shorter contexts (28K).

Context-length interaction pattern. The relative performance of the two methods varies systematically with context length. Answerability-gain shows advantages at longer contexts (112K, 224K) where credit assignment is more challenging, but underperforms at shorter contexts (28K, 56K) where outcome-only rewards suffice. This suggests the dense reward signal may be more beneficial when evidence is harder to locate, but introduces noise when the task is simpler.

Table 2: Head-to-head comparison of Answerability-Gain vs. Outcome-Only across all conditions. Win/Loss determined by EM difference. The proposed method shows no consistent advantage.

Condition	Outcome-Only	Ans.-Gain	Δ EM	Winner
Top-20% 28K	67.19	62.50	−4.69	Outcome-Only
Top-20% 56K	68.75	66.41	−2.34	Outcome-Only
Top-20% 112K	64.06	66.41	+2.35	**Ans.-Gain**
Top-20% 224K	61.72	60.94	−0.78	Outcome-Only
Standard 28K	67.19	63.28	−3.91	Outcome-Only
Standard 56K	63.28	64.06	+0.78	**Ans.-Gain**
Standard 112K	60.94	62.50	+1.56	**Ans.-Gain**
Standard 224K	54.69	59.38	+4.69	**Ans.-Gain**
Overall Avg	**63.48**	**63.19**	**−0.29**	**Outcome-Only**

Table 3: Training dynamics showing gate behavior evolution. Update rate indicates fraction of chunks where memory is updated; exit position indicates relative position in context where model exits (0=beginning, 1=end).

Method	Step	Update Rate	Exit Position	Num Steps
Outcome-Only	1	0.906	0.419	2.43
Outcome-Only	15	0.952	0.681	3.91
Outcome-Only	25	0.927	0.630	3.65
Outcome-Only	36	0.923	0.617	3.50
Answerability-Gain	1	0.912	0.412	2.38
Answerability-Gain	9	0.876	0.607	3.52
Answerability-Gain	15	0.889	0.601	3.45
Answerability-Gain	21	0.941	0.672	3.89

4.3 Head-to-Head Comparison

Table 2 provides a direct comparison between answerability-gain and outcome-only across all eight evaluation conditions.

The results show a 4–4 win/loss split between the two methods. Answerability-gain wins at longer contexts (112K Top-20%, 56K–224K Standard) while outcome-only wins at shorter contexts (28K–56K Top-20%, 28K Standard). The overall average EM favors outcome-only by 0.29 percentage points (63.48% vs. 63.19%), indicating that the proposed dense reward signal does not provide a net benefit across the evaluation suite.

4.4 Training Dynamics

To understand the learned gating behavior, we analyze training dynamics in Table 3.

Update gate exhibits near-degenerate behavior. Under both reward schemes, the update rate remains high throughout training (87–95%), indicating that the update gate does not learn selective memory updating. Instead, the model updates memory on nearly every chunk it processes, regardless of whether the chunk contains evidence.

Exit gate learns meaningful patterns. In contrast, the exit position increases during training (from ∼0.41 to ∼0.62–0.67), indicating that both methods learn to read more of the context before exiting. The exit gate is the primary mechanism for efficiency gains, while the update gate contributes little to selectivity.

Both methods converge to similar behavior. Despite the different reward signals, both methods converge to similar gate behavior patterns: high update rates and moderate exit positions. This suggests that the answerability-gain signal does not fundamentally change how the gates learn to operate.

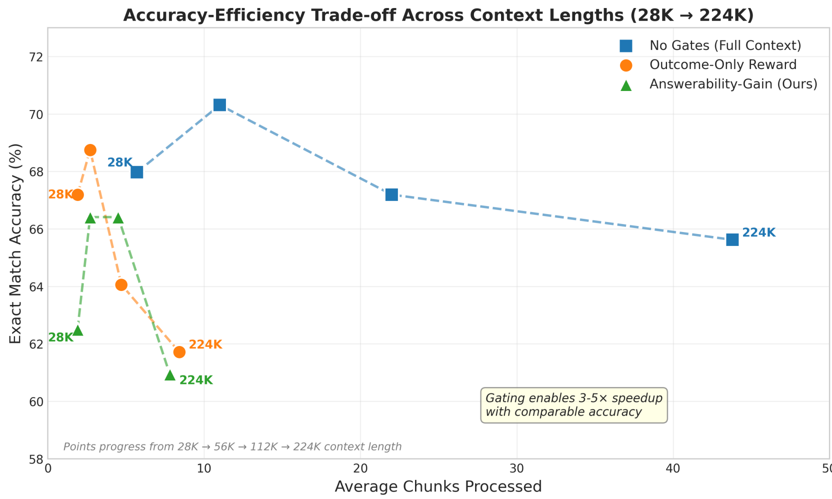

Figure 2: Accuracy-efficiency trade-off across context lengths (28K–224K tokens). Gated methods (orange, green) achieve comparable accuracy to full-context processing (blue) while using 3–5× fewer chunks. Points progress from shorter to longer contexts along each curve.

4.5 Efficiency Trade-off

Figure 2 visualizes the accuracy-efficiency trade-off across all methods and context lengths.

The scatter plot confirms that both gated methods occupy similar positions on the accuracy-efficiency frontier. Neither method consistently dominates the other: outcome-only achieves slightly higher accuracy at lower chunk counts (shorter contexts), while answerability-gain shows marginal advantages at higher chunk counts (longer contexts). The no-gates baseline achieves the highest accuracy but at substantially higher computational cost.

4.6 Analysis: Why Answerability-Gain Underperforms

Our experiments reveal that answerability-gain rewards do not consistently improve over outcome-only rewards for gate training. We identify an architectural limitation that explains this finding.

Early-exit bias. The answerability-gain signal exhibits a bias toward early exit after encountering the first piece of evidence. When the model processes a chunk containing evidence, the answer confidence increases substantially, producing a high positive gain. This encourages the exit gate to terminate before gathering additional evidence that may be needed for multi-hop reasoning.

Evidence for the bias. Several observations support this hypothesis: (1) Answerability-gain performs relatively better on Standard placement at longer contexts, where early exit is less harmful because evidence is distributed throughout the document. (2) The update gate remains near-degenerate under both reward schemes, suggesting that the gain signal primarily influences exit decisions rather than memory update decisions. (3) Two rounds of hyperparameter optimization (reward rebalancing, advantage normalization) failed to improve results, indicating the issue is architectural rather than a tuning problem.

Implications. Dense rewards based on answer confidence may not be suitable for tasks requiring evidence integration from multiple sources. Alternative approaches might include multi-step lookahead rewards that consider future evidence, evidence-diversity bonuses that encourage gathering varied information, or curriculum learning that progresses from single-hop to multi-hop questions.

5 Conclusion

We investigated answerability-gain rewards as a dense, evidence-label-free signal for training GRU-Mem gating mechanisms. Despite the theoretical appeal of rewarding memory updates based on answer confidence improvements, our comprehensive experiments on RULER-QA (28K–224K tokens) reveal that this approach does not consistently outperform simpler outcome-only rewards, achieving 63.19% vs. 63.48% average EM with a 4–4 win/loss split across conditions. We identify an architectural limitation: the gain signal biases toward early exit after encountering the first evidence, which hurts multi-hop reasoning tasks. Future work might explore multi-step lookahead rewards, evidence-diversity bonuses, or curriculum learning to address this limitation.

References

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

A. Bulatov, Yuri Kuratov, and M. Burtsev. Recurrent memory transformer. *ArXiv*, abs/2207.06881, 2022.

Alex J. Chan, Hao Sun, Samuel Holt, and M. Schaar. Dense reward for free in reinforcement learning from human feedback. *ArXiv*, abs/2402.00782, 2024.

Zihang Dai, Zhilin Yang, Yiming Yang, J. Carbonell, Quoc V. Le, and R. Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *ArXiv*, abs/1901.02860, 2019.

Alex Graves. Adaptive computation time for recurrent neural networks. *ArXiv*, abs/1603.08983, 2016.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *ArXiv*, abs/2404.06654, 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. *ArXiv*, abs/2106.09685, 2021.

Zi-Yan Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *ArXiv*, abs/2503.20783, 2025.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *ArXiv*, abs/2207.07061, 2022.

Zhihong Shao, Peiyi Wang, Qihao Zhu, R. Xu, Jun-Mei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv*, abs/2402.03300, 2024.

Leheng Sheng, Yongtao Zhang, Wenchang Ma, Yaorui Shi, Ting Huang, Xiang Wang, An Zhang, Ke Shen, and Tat-Seng Chua. When to memorize and when to stop: Gated recurrent memory for long-context reasoning, 2026. URL `https://arxiv.org/abs/2602.10560`.

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information gain-based policy optimization: A simple and effective approach for multi-turn llm agents. *ArXiv*, abs/2510.14967, 2025.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, R. Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. pp. 2369–2380, 2018.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent, 2025. URL https://arxiv.org/abs/2507.02259.

Ziqian Zeng, Yihuai Hong, Hongliang Dai, Huiping Zhuang, and Cen Chen. Consistentee: A consistent and hardness-guided early exiting method for accelerating language models inference. *ArXiv*, abs/2312.11882, 2023.