Why Limit the Residual Stream to Layers and Not Tokens? Persistent Memory for Continuous Latent Reasoning

Mujtaba Farhan [1] **Maheep Chaudhary** [2]

Abstract

Large language models (LLMs) have demonstrated remarkable reasoning abilities on mathematical and multi-hop planning tasks. The CoCoNuT (Chain of Continuous Thought) paradigm (Hao et al., 2024) extends this by enabling models to reason in latent space, exploring multiple reasoning paths simultaneously rather than committing to a single chain early on. However, we identify a limitation we term the **concept bottleneck**. At each reasoning pass, intermediate hidden states are overwritten, causing the model to lose critical facts computed in earlier steps as reasoning depth increases. We observe this empirically. On HotpotQA, vanilla CoCoNuT (10.4% EM) fails to improve over the CoT baseline (11.0% EM), and performance degrades with curriculum depth on GSM8K. To address this, we propose **AGCLR** (Adaptive Gated Continuous Latent Reasoning), which augments CoCoNuT with a *Gated Concept Stream*. A persistent residual memory maintained across all reasoning passes, controlled by three learned gates: a *write* gate that commits intermediate facts to memory, a *read* gate that retrieves relevant prior states, and a *forget* gate that prunes irrelevant context. Evaluated on GSM8K, HotpotQA, and ProsQA using GPT-2 as our base model, AGCLR achieves consistent improvements across all types of datasets. With the performance gap compounding as curriculum depth increases, directly resolving the concept bottleneck. Code available at `https://anonymous.4open.science/r/JJJJ/README.md`

[1]Algoverse AI Research [2]Independent. Correspondence to: Anonymous <anon@anonymous.edu>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

1. Introduction

Multi-step reasoning remains one of the most challenging aspects of large language model capabilities. Wei et al. (2022) showed that prompting LLMs with intermediate reasoning steps significantly improves performance on mathematical and logical benchmarks. However, Chain-of-Thought (CoT) reasoning is constrained to a single forward pass. Each token generated becomes the input for the next, forcing the model to commit to a reasoning path early and preventing the exploration of alternative paths. Moreover, explicit reasoning traces are often incomplete or unfaithful to the underlying computation (Su et al., 2026; Swaroop et al., 2025), motivating reasoning that operates directly in latent space. More recent work has explored internalizing these

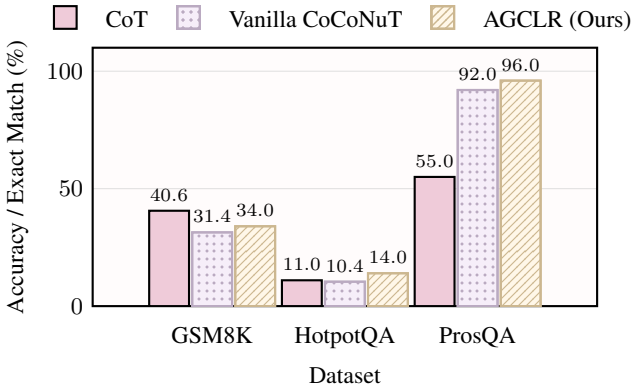

Figure 1. **AGCLR excels at multi-hop reasoning.** Performance across GSM8K (math), HotpotQA (multi-hop QA), and ProsQA (planning). AGCLR's persistent memory enables strong gains on multi-hop tasks (HotpotQA: +3.6%, ProsQA: +4.0%), while CoT remains superior for single-step mathematical reasoning.

reasoning chains. Deng et al. (2024) proposed iCoT, which progressively removes the prefix of reasoning chains during training until the model predicts answers without any explicit chain. Goyal et al. (2023) introduced pause tokens, fixed-embedding special tokens inserted between question and answer to provide extra compute time. Both approaches operate in language space and cannot maintain persistent state across reasoning steps.

The most ambitious extension is CoCoNuT (Hao et al.,

2024), which replaces discrete reasoning tokens with continuous latent thoughts. The model's last hidden state is fed back directly as the next input embedding, enabling reasoning in an unconstrained latent space and supporting implicit breadth-first search over reasoning paths. CoCoNuT is trained via a multi-stage curriculum that progressively replaces explicit reasoning steps with latent tokens, one step per stage. Despite its promise, vanilla CoCoNuT suffers from a **concept bottleneck**: intermediate reasoning states are progressively lost across multi-pass inference, as each new latent token overwrites information from earlier passes with no persistent memory. This becomes severe in multi-hop reasoning requiring longer chains. We demonstrate this empirically across GSM8K (arithmetic), HotpotQA (multi-hop QA), and ProsQA (planning) in Figure 1.

To address this, we propose **AGCLR** (Adaptive Gated Continuous Latent Reasoning), which augments CoCoNuT with a gated concept stream that preserves intermediate reasoning states across passes. While gating mechanisms trace back to LSTMs (Hochreiter & Schmidhuber, 1997) for sequential state updates, our gates operate on *persistent cross-pass memory* in continuous latent reasoning: each pass refines the same representation rather than processing new sequential inputs, and memory accumulates facts across iterative reasoning cycles rather than discarding them at each timestep.

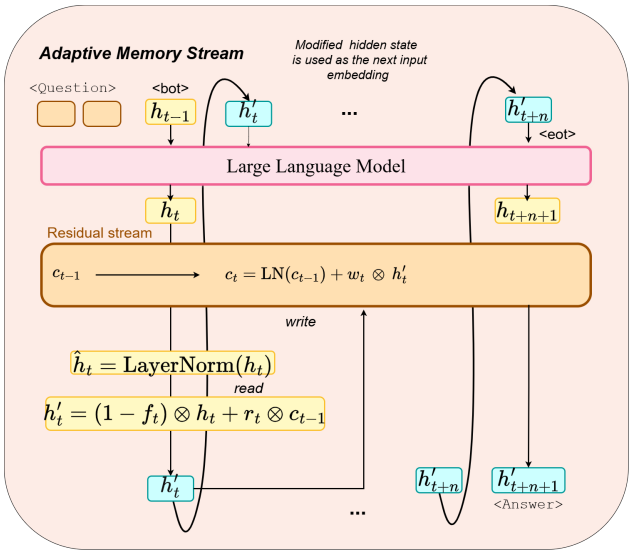

Figure 2. **AGCLR architecture.** At each latent token position, three learned gates (read, forget, write) control information flow between the current hidden state h_t and the persistent concept stream c_t. The read gate retrieves relevant prior facts from c_{t-1}, the forget gate prunes irrelevant context from h_t, and the write gate commits the gated hidden state h'_t to the residual stream, directly addressing the concept bottleneck in vanilla CoCoNuT.

AGCLR augments CoCoNuT with a *Gated Concept Stream*: a persistent residual memory vector $c_t \in \mathbb{R}^d$ maintained across all reasoning passes. At each latent token position, three learned sigmoid gates control information flow: a *write* gate commits relevant intermediate facts from the current hidden state to memory; a *read* gate retrieves prior memory into the current reasoning state; and a *forget* gate prunes irrelevant context from the hidden state. Figure 2 illustrates the architecture.

We make the following contributions:

- We identify and empirically demonstrate the **concept bottleneck** in vanilla CoCoNuT across three reasoning datasets of different types.
- We propose **AGCLR**, a gated residual memory mechanism that resolves the concept bottleneck with only 1.41% additional parameters over GPT-2.
- AGCLR consistently outperforms vanilla CoCoNuT on GSM8K (arithmetic), HotpotQA (multi-hop QA), and ProsQA (graph planning), with the advantage compounding as curriculum depth increases.

2. Related Work

Gating mechanisms for controlling information flow trace back to Long Short-Term Memory networks (Hochreiter & Schmidhuber, 1997), which introduced forget gates to selectively retain or discard information in recurrent hidden states. However, LSTMs gate sequential inputs across timesteps, whereas our gates operate on persistent memory across iterative reasoning passes over the same latent representation. (Deng et al., 2024) proposed iCoT, which progressively removes explicit reasoning prefix tokens during training; while iCoT compresses reasoning into the forward pass, it lacks any mechanism to preserve information across reasoning steps and does not operate in a multi-pass latent reasoning setting. (Hao et al., 2024) introduced CoCoNuT, which enables continuous latent reasoning by recursively feeding the model's hidden state back as the next input embedding, allowing implicit breadth-first search over reasoning paths. CoCoNuT serves as our direct baseline across all three datasets, but discards all prior hidden states at each pass and lacks persistent memory, leading to the concept bottleneck we identify and address. (Wang et al., 2024) proposed a concurrent post-training approach using a fixed scalar α to blend consecutive hidden states at inference time; unlike our method, their gates are not learned end-to-end, operate only on consecutive states rather than a persistent residual stream, and are applied training-free as post-processing. Relatedly, amortized latent steering learns a low-cost intervention that substitutes for test-time latent optimization (Egbuna et al., 2025); like the fixed-α blend, however, it applies a transient steering signal rather than maintaining a persistent, gated residual stream across passes. Memory-augmented architectures such as Neural

Turing Machines (Graves et al., 2014) and Differentiable Neural Computers (Graves et al., 2016) have explored external memory for sequential reasoning, but augment models with external read/write operations across sequence chunks rather than maintaining persistent internal state within multi-pass latent reasoning as we do.

3. Method: AGCLR

3.1. Gated Concept Stream

We augment CoCoNuT with a persistent concept stream $c_t \in \mathbb{R}^d$, initialized to zero at the start of each forward call and updated at every latent token position. At pass t, given hidden state h_t at the latent token position:

$$\hat{h}_t = \text{LayerNorm}(h_t), \tag{1}$$

$$r_t = \sigma(W_r \hat{h}_t), \quad f_t = \sigma(W_f \hat{h}_t), \quad w_t = \sigma(W_w \hat{h}_t), \tag{2}$$

$$h'_t = (1 - f_t) \odot h_t + r_t \odot c_{t-1}, \tag{3}$$

$$c_t = \text{LayerNorm}(c_{t-1} + w_t \odot h'_t), \tag{4}$$

where $r_t, f_t, w_t \in [0, 1]^d$ are the read, forget, and write gates, and $W_r, W_f, W_w \in \mathbb{R}^{d \times d}$ are learned weight matrices. The gated hidden state h'_t replaces h_t as the input embedding for the next latent token position.

Read gate r_t controls how much of the concept stream c_{t-1} is retrieved into the current hidden state, allowing pass t to access facts from all earlier passes. **Forget gate** f_t controls how much of the current hidden state is preserved versus replaced by retrieved memory, enabling selective pruning of irrelevant context. **Write gate** w_t controls how much of the gated hidden state h'_t is committed to the concept stream, preventing low-confidence states from polluting the residual memory.

4. Training Protocol

4.1. Multi-Stage Curriculum

We leverage language Chain-of-Thought data to supervise continuous latent reasoning by implementing a multi-stage training curriculum inspired by Hao et al. (2024). In the initial stage (Stage 0), the model is trained on regular CoT instances with explicit reasoning steps. In subsequent stages, we progressively replace reasoning steps with continuous latent thoughts. At stage k, the first k reasoning steps in the CoT are replaced with $k \times c$ latent tokens, where c is a hyperparameter controlling the number of latent thoughts replacing a single language reasoning step. We insert <bot> (beginning of thought) and <eot> (end of thought) tokens to encapsulate the continuous thoughts. Following Hao et al. (2024), we reset the optimizer state when transitioning between training stages.

4.2. Implementation Details

We use a pre-trained GPT-2 base model (117M parameters) with a learning rate of 1×10^{-4} and effective batch size of 128. We train on three multi-hop reasoning benchmarks: GSM8K (Cobbe et al., 2021), HotpotQA (Yang et al., 2018), and ProsQA (Hao et al., 2024). Following the curriculum structure from vanilla CoCoNuT (Hao et al., 2024), we progress through Stages 0–2 (partially latent reasoning) during epochs 1–9, incrementally replacing reasoning steps with latent tokens. From epoch 10 onwards, we remain in Stage 3 where all reasoning is latent, training for 15 total epochs on GSM8K and HotpotQA, and 20 epochs on ProsQA (which contains more complex reasoning chains with up to 6 steps). For HotpotQA, we format instances to include the question, supporting paragraphs, intermediate reasoning steps, and answer span to encourage multi-hop reasoning during CoT stages. The checkpoint with the best validation accuracy in the final stage is used for evaluation.

5. Results

5.1. Main Results

Table 1 shows AGCLR consistently outperforming vanilla CoCoNuT across all three datasets.

5.2. Alleviating the Concept Bottleneck

Vanilla CoCoNuT and AGCLR perform comparably at early curriculum stages, but AGCLR's advantage compounds as reasoning depth increases. On ProsQA, vanilla CoCoNuT peaks at 95% accuracy at stage 5 but **degrades to 92%** at stage 6—the final curriculum stage where all reasoning steps are replaced by latent tokens. AGCLR sustains improvement, achieving 96% at the same checkpoint. This degradation-vs-improvement pattern demonstrates the **concept bottleneck**: as models transition to fully latent reasoning, intermediate computational states are progressively lost without a preservation mechanism.

Memory Retention Across Reasoning Passes. To understand how gating resolves this bottleneck, we analyze hidden state evolution across reasoning passes. Figure 3 shows cosine similarity between pass-1 hidden states and subsequent passes, measured on 100 validation samples at epoch 15. Vanilla CoCoNuT exhibits monotonic memory decay: similarity drops from 1.0 to 0.126 by pass 6, representing 87% information loss as intermediate reasoning steps are progressively overwritten. **AGCLR mitigates this decay.** While similarity initially drops (pass 1→2), it stabilizes at ∼0.22 for passes 3–6, retaining 71% more information than vanilla CoCoNuT at final generation (0.216 vs 0.126). The gated concept stream acts as a persistent memory buffer, preserving critical reasoning state across passes. This memory preservation directly explains AGCLR's improvement

Method	GSM8K	HotpotQA		ProsQA
	Acc. (%)	EM (%)	F1 (%)	Acc. (%)
CoT (Wei et al., 2022)	40.6	11.0	15.5	55.0
No-CoT (Hao et al., 2024)	16.5	4.0	7.6	76.7
iCoT (Deng et al., 2024)†	30.0	6.6	9.4	98.2
Pause Token (Goyal et al., 2023)†	16.4	10.6	14.6	75.9
Vanilla CoCoNuT (Hao et al., 2024)	31.4	10.4	15.2	92.0
AGCLR (Ours)	**34.0**$_{+2.6}$	**14.0**$_{+3.6}$	**19.4**$_{+4.2}$	**96.0**$_{+4.0}$

Table 1. **Results on three datasets: GSM8K, HotpotQA and ProsQA.** Higher accuracy indicates stronger reasoning. †Results from Deng et al. (2024) using identical GPT-2 architecture, as reported in Hao et al. (2024). HotpotQA not evaluated in prior work. ProsQA evaluated at stage 6 (all reasoning steps latent) for fair comparison.

+3.6% EM on HotpotQA.

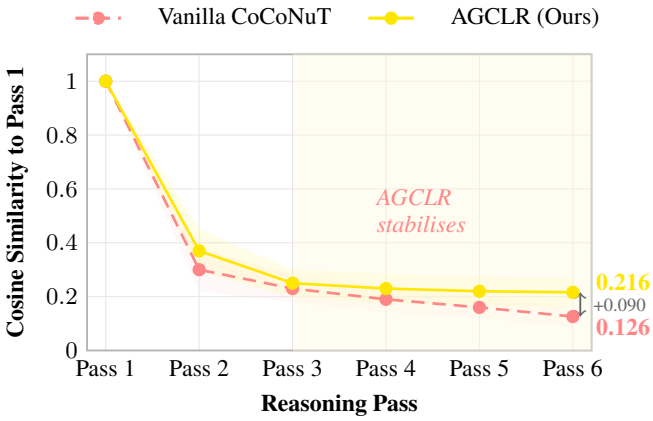

Figure 3. **Hidden State Memory Retention.** Cosine similarity between pass-1 and subsequent passes (100 samples, epoch 15). Vanilla CoCoNuT exhibits monotonic decay (1.0→0.126), while AGCLR stabilizes after pass 3. Shaded regions: ±1 std. AGCLR retains 71% more information (+0.090 gap at pass 6), enabling +4.0% gains on ProsQA.

5.3. What Gets Written to the Concept Stream

To understand how AGCLR preserves task-relevant information, we analyze concept stream content by computing cosine similarities between its embeddings and vocabulary tokens. Figure 4 visualizes three contrastive examples where AGCLR answers correctly but vanilla CoCoNuT fails. Answer components consistently achieve high similarity scores (0.5–0.8, darker regions in heatmap): ”Penn” (0.806) and ”William” (0.681) for the Manor Township question, ”War” (0.760) and ”World” (0.684) for the Oakland Assembly question, and ”Pennsylvania” (0.479) with abstraction markers ”Federal” (0.507) for the WCDL question. These patterns demonstrate that the gated concept stream stores distributed representations of entities (0.6–0.8 similarity) and their semantic associations (0.4–0.5 similarity).This preservation mechanism directly explains AGCLR’s performance advantage. In the Manor Township example, AGCLR maintains the Pennsylvania→William Penn binding needed for correct generation, while vanilla CoCoNuT hallucinates ”Henry David Thoreau” after losing this entity relationship across passes. Similarly, the Oakland Assembly case shows temporal context preservation (”War”/”World” similarities prevent drift to ”World War II”), and the WCDL example demonstrates multi-hop reasoning where both the base entity (”Pennsylvania”) and abstraction markers (”Federal”/”country”) enable geographic generalization. These findings directly explain the 71% better information retention measured in Figure 3 and AGCLR’s +3.6% EM improvement over vanilla CoCoNuT.

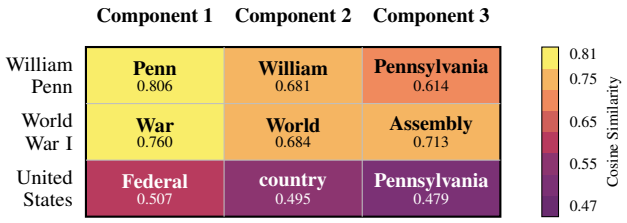

Figure 4. **Concept stream entity preservation heatmap.** Cosine similarities between concept stream embeddings and answer-relevant tokens for three examples where AGCLR succeeds and vanilla CoCoNuT fails. AGCLR preserves the Pennsylvania→William Penn binding (0.81), temporal context for World War I (0.76), and geographic abstraction for United States (0.51), consistently achieving 0.6–0.8 similarity for task-relevant entities, explaining the +3.6% EM gain over vanilla CoCoNuT.

6. Conclusion

We identified the concept bottleneck—where intermediate states are lost across passes—and proposed AGCLR with a gated concept stream to resolve it. AGCLR outperforms vanilla CoCoNuT across all three datasets, with gains compounding at deeper stages. On ProsQA, AGCLR reaches 96% accuracy while vanilla degrades to 92%, directly demonstrating that persistent memory resolves reasoning depth limitations. Ablations show gains stem from persistent memory, not parameters: freezing write after pass 2 yields only -0.8% EM loss, confirming early capture suf-

fices. Our evaluation uses single-seed GPT-2 124M on three benchmarks; while consistent improvements across diverse reasoning tasks suggest the approach generalizes, scalability to larger models and broader task distributions remains to be validated. Future work includes scaling to larger base models and multi-seed evaluation.

References

Chaudhary, M. and Geiger, A. Evaluating open-source sparse autoencoders on disentangling factual knowledge in gpt-2 small. *arXiv preprint arXiv:2409.04478*, 2024.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Deng, Y., Choi, Y., and Shieber, S. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*, 2024.

Egbuna, N., Gaur, S., Dev, S., Panda, A., and Chaudhary, M. Amortized latent steering: Low-cost alternative to test-time optimization. *arXiv preprint arXiv:2509.18116*, 2025.

Golechha, S., Chaudhary, M., Velja, J., Abate, A., and Schoots, N. Modular training of neural networks aids interpretability. *arXiv e-prints*, pp. arXiv–2502, 2025.

Goyal, S., Ji, Z., Rawat, A. S., Menon, A. K., Kumar, S., and Nagarajan, V. Think before you speak: Training language models with pause tokens. *arXiv preprint arXiv:2310.02226*, 2023.

Graves, A., Wayne, G., and Danihelka, I. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwinska, A., Colmenarejo, S. G., Grefenstette, E., Ramalho, T., Agapiou, J., et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.

Hao, S., Sukhbaatar, S., Su, D., Li, X., Hu, Z., Weston, J., and Tian, Y. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Su, I., Purushothaman, G., Narayan, J., Goel, R., Zhu, K., Dev, S., More, Y., and Chaudhary, M. Broken chains: The cost of incomplete reasoning in llms. *arXiv preprint arXiv:2602.14444*, 2026.

Swaroop, A., Nallani, A., Uboweja, S., Uzdenova, A., Nguyen, M., Zhu, K., Dev, S., Panda, A., Sharma, V., and Chaudhary, M. Frit: Using causal importance to improve chain-of-thought faithfulness. *arXiv preprint arXiv:2509.13334*, 2025.

Wang, X., Wang, D., Ying, W., Bai, H., Gong, N., Dong, S., Liu, K., and Fu, Y. Efficient post-training refinement of latent reasoning in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33692–33700, 2024.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, 2022.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

A. Extended Ablation Studies

A.1. Persistent Memory vs Additional Parameters

To validate that AGCLR's performance gains stem from *persistent memory mechanisms* rather than simply additional parameters, we conduct an ablation study examining the role of dynamic writing across reasoning passes. We compare four configurations: (1) Vanilla CoCoNuT baseline, (2) AGCLR without write gate (read and forget only), (3) AGCLR with write gate frozen after pass 2, and (4) Full AGCLR with all gates active.

The key question is whether the write gate's value lies in *early information capture* (passes 1–2) or *continuous refinement* across all passes. If dynamic writing throughout all passes were critical, we would expect significant performance degradation when the write gate is frozen. Conversely, if early capture combined with persistent retrieval is sufficient, performance should remain largely intact.

Figure 5 presents our results. Remarkably, freezing the write gate after pass 2 results in only minimal performance degradation: 13.2% EM versus 14.0% EM for full AGCLR (-0.8% absolute). This near-equivalent performance demonstrates that early information capture is largely sufficient for multi-hop reasoning. Notably, the model without any write gate achieves only 8.8% EM, confirming that the write gate is necessary—but its primary value lies in the initial passes. The read and forget gates then maintain and retrieve this early-captured information throughout subsequent reasoning steps.

This finding reveals AGCLR's core mechanism: the concept stream functions as *persistent storage* rather than a dynamic scratchpad. Information is written once during early passes (1–2), then read and selectively forgotten across later passes (3–6). The write gate's primary value lies in identifying and capturing relevant information early, not in continuously refining the stored representation. This validates our hypothesis that AGCLR's gains emerge from the persistent memory architecture itself—the ability to store, maintain, and retrieve information across reasoning steps—rather than from simply adding more trainable parameters to the model.

A.2. Individual Gate Ablations

We systematically remove each gate by fixing its output to zero throughout training to understand their individual contributions. Results are presented in Table 2.

Removing any single gate degrades performance, confirming all three components are necessary. The write gate is most structurally critical: without it, nothing is committed to the concept stream, dropping EM to 8.8%, below vanilla CoCoNuT (10.4%). However, the forget gate produces the *largest performance drop* (8.4% EM, -5.6%), demonstrating that selective forgetting is essential for maintaining concept stream quality. In multi-hop reasoning, intermediate computations accumulate both relevant facts (e.g., entity names needed for later hops) and irrelevant context (e.g., formatting tokens, partial calculations from earlier steps). Without the forget gate pruning this noise, the concept stream becomes polluted across passes, degrading retrieval quality and preventing the model from isolating task-relevant information for final answer generation. Removing the read gate costs 4.6% EM, confirming that cross-pass retrieval drives a meaningful share of AGCLR's gains. All three gates are indispensable to resolving the concept bottleneck.

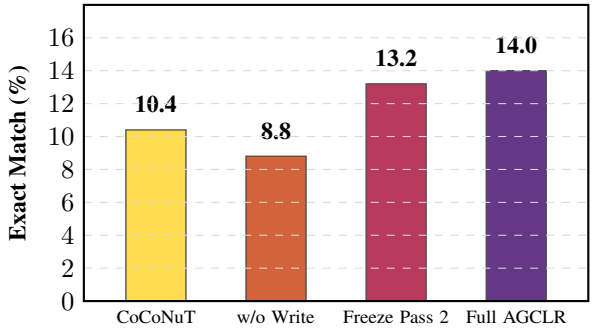

Figure 5. **Ablation study of gating mechanisms.** Exact Match on HotpotQA. *CoCoNuT*: vanilla baseline (10.4%). *w/o Write*: write gate removed (8.8%). *Freeze Pass 2*: write gate frozen after pass 2 (13.2%). *Full AGCLR*: all gates active (14.0%). The small gap between Freeze Pass 2 and Full AGCLR ($+0.8\%$ EM) shows early capture suffices; the large drop without the write gate confirms it is critical.

Method	HotpotQA EM (%)	HotpotQA F1 (%)
AGCLR (full)	14.0	19.4
w/o write gate	8.8	17.1
w/o read gate	9.4	17.8
w/o forget gate	8.4	18.9
Vanilla CoCoNuT	10.4	15.2

Table 2. **Individual gate ablation results.** Each gate is removed by fixing its output to zero throughout training.

A.3. Initialization

Gate weights W_r, W_f, W_w are initialized to zero for stable warm-up. Gate biases use dataset-specific values (Table 3): lower forget/higher write for ProsQA (preserves graph entities), higher forget/lower write for GSM8K (prunes intermediate steps).

This conservative initialization strategy prevents premature concept stream saturation. The low write gate activation (0.18) encourages the model to be selective about what information gets stored initially. The moderate forget gate activation (0.27) retains most information early in training,

Dataset	Read	Forget	Write
GSM8K / HotpotQA	0.43	0.27	0.18
ProsQA	0.43	0.18	0.43

Table 3. Dataset-adaptive gate initialization values $\sigma(b)$.

allowing the model to learn which facts are relevant. The read gate's balanced initialization (0.43) provides moderate retrieval strength. During training, gates adapt to dataset-specific patterns (Table 2), demonstrating that the architecture learns task-appropriate memory management rather than relying on fixed heuristics.

A.4. Concept Stream Content Analysis: Detailed Examples

Table 4 provides detailed examples of what information gets written to the concept stream, showing cosine similarities between concept stream embeddings and vocabulary tokens for three contrastive cases where AGCLR answers correctly but vanilla CoCoNuT fails.

These examples demonstrate three key patterns: (1) Entity binding preservation—AGCLR maintains the Pennsylvania→William Penn relationship while vanilla CoCoNuT loses it and hallucinates a plausible but incorrect historical figure. (2) Temporal context retention—high similarity to "War" and "World" prevents drift to the more statistically common "World War II" after losing specific temporal markers. (3) Multi-hop reasoning—the concept stream preserves both base entities ("Pennsylvania") and abstraction markers ("Federal", "country"), enabling geographic generalization from state to country level, while vanilla CoCoNuT stops at the first hop.

B. Implementation Details

B.1. Training Configuration

All models were trained using AdamW optimizer with learning rate 5×10^{-5}, weight decay 0.01, and batch size 128 (8 per device $\times$ 16 gradient accumulation steps on A10 GPU). We used the curriculum learning schedule from Hao et al. (2024), progressively increasing latent tokens from 0 to 6 across stages. Gate parameters were initialized following Meta's protocol: read gate bias at -0.28 ($\sigma^{-1}(0.43)$), forget gate at -1.0 ($\sigma^{-1}(0.27)$), and write gate at -1.5 ($\sigma^{-1}(0.18)$).

B.2. Evaluation Details

Exact Match (EM) and F1 scores were computed following standard QA evaluation protocols. For HotpotQA, we evaluated on 500 randomly sampled validation examples at stage 3 (6 latent tokens). For ProsQA, we report accuracy on the full validation set. All cosine similarity analyses used 100 validation samples at epoch 15, averaging across passes and examples.

B.3. Computational Requirements

AGCLR training on HotpotQA required approximately 15 epochs ($\sim$2 hours on a single NVIDIA A10 GPU). ProsQA training reached 96% accuracy in 20 epochs on a single GH200 GPU. Total training time across all three datasets (GSM8K, HotpotQA, ProsQA) was approximately 8 hours on a single A10 GPU.

C. Additional Limitations

Beyond the limitations discussed in the main paper, we note several additional considerations:

Single-seed evaluation. All reported results are from single training runs. While consistent improvements across three diverse datasets suggest robustness, variance estimates from multiple seeds would strengthen claims.

Model scale. Our experiments use GPT-2 124M as the base model. Scalability to larger models (1B+ parameters) remains unexplored, though the architectural modifications are parameter-agnostic.

Dataset coverage. While we evaluate on arithmetic (GSM8K), multi-hop QA (HotpotQA), and planning (ProsQA), additional benchmarks (MuSiQue, 2WikiMultihopQA, StrategyQA) would further validate generalization.

Comparison to retrieval systems. We focus on controlled architectural comparisons against vanilla CoCoNuT. Comparison to retrieval-augmented systems or significantly larger models (e.g., GPT-3.5, GPT-4) is beyond our scope but would provide additional context.

Interpretability depth. Our concept-stream analysis follows a logit-lens-style probing; finer-grained feature attribution (Chaudhary & Geiger, 2024; Golechha et al., 2025) is left to future work.

Question	AGCLR Answer	Vanilla CoCoNuT Answer	Concept Stream Similarities
Who founded Manor Township, Pennsylvania?	William Penn	Henry David Thoreau	Penn: 0.806 William: 0.681 Pennsylvania: 0.614
When did Oakland Assembly close?	World War I	World War II	War: 0.760 World: 0.684
What country is WCDL radio station in?	United States	Pennsylvania	Pennsylvania: 0.479 Federal: 0.507 country: 0.495

Table 4. **Concept stream content examples.** Cosine similarities between concept stream embeddings and vocabulary tokens for three contrastive examples. High similarities (0.5–0.8) indicate successful storage of answer-relevant entities and their semantic associations.