Does COCONUT Reason or Buffer? Dissecting Latent Thought Tokens on ProsQA

Brian Martin Independent

Stephen Lipmann Independent

Abstract

Meta's COCONUT replaces explicit chain-of-thought with continuous latent thought tokens, recycling transformer hidden states as input embeddings across multiple reasoning steps. On ProsQA, a synthetic graph-traversal benchmark, COCONUT achieves 97% accuracy, substantially outperforming chain-of-thought baselines. The original work attributes this gain to the continuous latent space, which is hypothesized to encode richer, parallelized reasoning than discrete token sequences. However, COCONUT's training relies on a 7-stage curriculum that progressively removes explicit reasoning tokens, a confound that has not been controlled for. We construct M5, a compute-matched pause-token baseline that receives the same GPT-2 124M architecture, the same curriculum schedule, and the same number of latent positions, but replaces the recycled hidden states with fixed learned pause embeddings that carry no information between steps. M5 reaches 95.6% test accuracy, closing 85% of the gap between chain-of-thought (83.0%) and COCONUT (98.0%). Three converging experiments fail to distinguish the two systems: token corruption produces identical degradation profiles with zero permutation sensitivity, linear probes recover comparable intermediate reasoning information (peak accuracy 55.4% for COCONUT vs. 57.1% for M5, selectivity 0.0 for both), and cross-model thought transplantation succeeds in both directions. On out-of-distribution generalization, M5 outperforms COCONUT on 3 of 4 test sets, including a 9.4 percentage-point advantage on 7-hop paths ($p = 0.007$, Bonferroni-corrected). These results indicate that the curriculum, not the continuous latent mechanism, drives COCONUT's in-distribution performance, and that the recycling mechanism may constrain generalization.

---

1. Introduction

Chain-of-thought prompting demonstrates that large language models solve multi-step reasoning problems more reliably when they externalize intermediate steps as natural language tokens (Wei et al., 2022). This observation has motivated a line of work that asks whether explicit verbalization is necessary, or whether models can perform equivalent computation in a latent space without producing human-readable traces. COCONUT (Hao et al., 2024) offers the most direct test of this question: it trains a language model to replace chain-of-thought tokens with continuous thought tokens, recycling the transformer's final-layer hidden state back into the input embedding stream across multiple reasoning positions. On ProsQA, a synthetic graph-traversal task requiring multi-hop path finding, COCONUT achieves 97% accuracy, a 17-point improvement over its chain-of-thought baseline. The authors attribute this gain to the expressiveness of the continuous latent space, which they argue encodes a breadth-first search strategy that discrete tokens cannot represent.

This attribution faces an uncontrolled confound. COCONUT is trained with a 7-stage curriculum that progressively removes explicit reasoning tokens, forcing the model to internalize computation that was previously externalized. The curriculum transforms the training distribution, the loss landscape, and the model's learned representations simultaneously with the introduction of the recycling mechanism. Any performance

gain could arise from the curriculum alone, from the mechanism alone, or from their interaction. Without a control that isolates one factor from the other, the causal claim remains underdetermined.

We introduce M5, a pause-token baseline designed to resolve this confound. M5 shares every architectural and training detail with the COCONUT model (M3): the same GPT-2 124M backbone, the same 7-stage curriculum schedule, and the same number of latent thought positions per example. The single difference is that M5 replaces the recycled hidden-state embeddings with fixed learned pause vectors (Goyal et al., 2024). These pause tokens occupy the same computational budget but carry no information from one reasoning step to the next. If the continuous latent mechanism is causally responsible for COCONUT's performance, M5 should fail where M3 succeeds. If the curriculum is the primary driver, M5 should perform comparably.

M5 reaches 95.6% test accuracy, closing 85% of the gap between chain-of-thought (83.0%) and COCONUT (98.0%). Three additional experiments converge on the same conclusion. Corrupting thought tokens produces identical degradation profiles for both models, with zero sensitivity to permutation order. Linear probes trained on intermediate representations recover comparable information (peak accuracy 55.4% for M3 vs. 57.1% for M5), with selectivity scores of 0.0 indicating that probing accuracy does not exceed what a control achieves on random targets. Cross-model thought transplantation succeeds bidirectionally, confirming that neither model's latent representations carry privileged information. On out-of-distribution test sets, M5 outperforms M3 on 7-hop paths by 9.4 percentage points ($p = 0.007$, Bonferroni-corrected), on 8-hop paths by 7.6 points, and on dense graphs by 7.2 points, while M3 holds a non-significant 7.3-point advantage on DAG structures. The recycling mechanism does not help generalization; it appears to hinder it.

This paper makes three contributions. First, we introduce a compute-matched control methodology that isolates the curriculum from the mechanism in latent-reasoning systems, applicable beyond COCONUT to any architecture that claims gains from a training-time intervention confounded with a progressive curriculum. Second, we provide converging evidence from three independent experimental paradigms — corruption analysis, representational probing, and out-of-distribution generalization — that the continuous latent mechanism is not the causal source of COCONUT's in-distribution performance. Third, we show that the recycling mechanism constrains out-of-distribution generalization relative to the simpler pause baseline, suggesting that architectural complexity can reduce robustness when the training curriculum already provides the necessary inductive bias.

2. Related Work

2.1 Chain-of-Thought and Latent Reasoning

Wei et al. (2022) established that prompting large language models to produce intermediate reasoning steps substantially improves performance on arithmetic, commonsense, and symbolic tasks. This finding raised a natural question: is the verbalization itself necessary, or does the benefit come from the additional forward passes that intermediate tokens provide? Several architectures have since attempted to move reasoning into latent space, replacing human-readable traces with learned internal representations. COCONUT is the most prominent of these, but the question generalizes to any system that trades explicit reasoning for implicit computation.

2.2 COCONUT and Continuous Thought

Hao et al. (2024) proposed COCONUT, which replaces chain-of-thought tokens with continuous thought tokens by recycling the transformer's last-hidden-state output back into the embedding stream. Trained with a multi-stage curriculum on ProsQA, COCONUT achieves 97% accuracy and the authors argue that the contin-

uous space enables a breadth-first search strategy inaccessible to discrete tokens. Zhu et al. (2025) provided a theoretical foundation, proving that continuous thought tokens are strictly more expressive than discrete chain-of-thought under certain conditions. However, Zhang et al. (2025) challenged the empirical picture by applying causal interventions to COCONUT's latent tokens. They found that the continuous thoughts are largely causally inert: shuffling, zeroing, or replacing them with Gaussian noise produces minimal performance degradation. Our work complements Zhang et al. by constructing an explicit alternative — the pause baseline — that matches COCONUT's training regime while eliminating the recycling mechanism entirely.

2.3 Pause Tokens and Compute Buffering

Goyal et al. (2024) introduced pause tokens as a method for providing transformers with additional computation without requiring meaningful intermediate output. Appending learned, non-informative tokens to the input gives the model extra forward-pass steps, improving performance on tasks that benefit from additional depth. The pause-token framework provides a natural control for COCONUT: if the gains come from extra computation rather than from the content of the latent representations, a model trained with pause tokens under the same curriculum should perform comparably. Our M5 baseline instantiates this control.

2.4 Probing and Causal Analysis

We use two standard methods to interrogate internal representations. Linear probing (Ravichander et al., 2021) trains a linear classifier on frozen hidden states to measure whether a target variable is linearly decodable. Ravichander et al. demonstrated that high probing accuracy alone does not establish that a representation is used by the model, motivating the use of selectivity controls — probing on random targets to establish a baseline — which we adopt. For causal analysis, we draw on the intervention methodology of Meng et al. (2022), who developed ROME to localize factual associations in GPT by corrupting and restoring activations at specific layers and positions. We adapt this approach to thought-token positions, measuring whether corrupting latent representations produces differential degradation between COCONUT and the pause baseline. Our corruption experiments extend Zhang et al. (2025) by comparing two matched models rather than analyzing a single model in isolation.

3. Methods

3.1 Task: ProsQA

ProsQA is a synthetic graph-traversal benchmark introduced by Hao et al. (2024) to evaluate multi-hop reasoning. Each sample presents a set of inheritance rules over nonsense entities (e.g., "Alex is a jompus. Every jompus is a zhorpus. Every zhorpus is a brimpus."), followed by a two-choice question ("Is Alex a brimpus or a daxil?") whose answer requires traversing the implied entity graph from the named individual to one of the two candidate types. Graphs are trees with path lengths of 3 to 6 hops. The vocabulary comprises 38 species names and 17 person names. The dataset contains 17,886 training samples, 300 validation samples, and 500 test samples, all generated from the same distributional family.

ProsQA is the task on which COCONUT achieves its strongest reported results (~97% accuracy), substantially above chain-of-thought baselines (~80%). If the continuous thought mechanism provides a genuine reasoning advantage, this task is where that advantage should be most apparent. We therefore treat ProsQA as the strongest-case evaluation domain for the mechanism.

3.2 Models

We train three models, all initialized from the same pretrained GPT-2 124M checkpoint (Radford et al., 2019; `openai-community/gpt2`, 124M parameters, 12 transformer layers, 768-dimensional hidden states). Table 1 summarizes the model configurations.

Table 1. Model configurations. All share the same pretrained initialization, optimizer, and hyperparameters. M3 and M5 share the same curriculum schedule.

Model	Thought mechanism	Curriculum
M1 (CoT)	None – explicit text reasoning tokens	No stages (standard supervised)
M3 (COCONUT)	Hidden states from the previous forward pass recycled as input embeddings	7-stage progressive CoT removal
M5 (Pause)	Fixed learned pause embedding (`nn.Parameter`) at each thought position	Same 7-stage curriculum as M3

M5 is the critical control. It isolates the contribution of the continuous thought mechanism by holding all other factors constant: same pretrained initialization, same AdamW optimizer (lr = 1e-4, weight_decay = 0.01), same curriculum schedule (epochs_per_stage = 5, max_latent_stage = 6), same effective batch size of 128, and the same number of attention positions occupied by thought tokens during both training and inference. The sole difference is what occupies those positions: M3 recycles hidden states across multiple forward passes, creating a sequential information pathway between thought steps, while M5 uses a single learned embedding vector and runs a single forward pass, providing the same number of additional attention positions without any inter-step information flow.

We implemented M5 by adding a `feedback_mode` parameter to the `Coconut` class in Meta's official codebase (`coconut.py`). When `feedback_mode="continuous"` (default), the model operates as standard COCONUT (M3). When `feedback_mode="pause_curriculum"`, thought positions receive a learned `nn.Parameter` embedding and inference executes a single forward pass rather than the multi-pass recurrence loop. The total modification to Meta's codebase comprises: (1) the `feedback_mode` parameter and associated branching logic in `coconut.py`, (2) two lines in `run.py` to read `feedback_mode` from the YAML configuration and pass it to the model constructor, and (3) a new configuration file (`prosqa_m5_pause.yaml`) identical to the COCONUT configuration except for `feedback_mode: pause_curriculum`. No changes were made to `dataset.py` or `utils.py`.

3.3 Training

All models were trained for 50 epochs on the ProsQA training set (17,886 samples) using AdamW (lr = 1e-4, weight_decay = 0.01) with an effective batch size of 128 (batch size 32, gradient accumulation over 4 steps on a single GPU, matching Meta's original 4-GPU configuration of batch size 32 with no gradient accumulation). Training used fp32 precision, seed 0, and the optimizer was reset at the start of each epoch, following Meta's training protocol (`reset_optimizer: True`).

For the curriculum models (M3 and M5), training proceeds through 7 stages. Stage 0 (epochs 0–4) trains with full explicit chain-of-thought supervision. At each subsequent stage k (epochs 5k through 5k + 4), the last k reasoning steps in the CoT are replaced with thought tokens – continuous hidden states for M3 and fixed pause embeddings for M5. By stage 6 (epochs 30–49), all reasoning steps are latent, and the model receives only the problem statement and thought positions before generating its answer. Thought positions

are padded to the maximum count (`pad_latent_to_max: True`), yielding 6 thought positions per sample regardless of the underlying path length.

All training was conducted on a single NVIDIA H100 80GB GPU. M1 required approximately 8 hours; M3 and M5 each required approximately 28 hours due to the multi-pass forward loop (M3) and the longer sequences with thought tokens (both M3 and M5).

3.4 Experiments

We design three experiments, each probing a different aspect of the distinction between sequential latent reasoning and unstructured compute buffering. All experiments use the 500-sample ProsQA test set unless otherwise noted.

Experiment 1: Corruption Ablation. This experiment tests whether thought tokens encode a sequential reasoning chain or serve as an unordered compute buffer. We apply six corruption conditions to the thought token hidden states of both M3 and M5:

- *Forward corruption:* progressively replace thought positions 0, 0:1, 0:2, …, 0:5 with random embeddings drawn from a distribution matched to the model's actual thought token statistics.
- *Reverse corruption:* the same procedure applied from the final position backward.
- *Single-position corruption:* replace only position k for each k in {0, …, 5}.
- *Permutation:* shuffle the order of the model's own thought token hidden states for the same problem (10 random permutations per sample, 500 samples). If thought tokens encode a sequential chain, reordering should degrade accuracy.
- *Partial permutation:* swap only adjacent pairs of thought tokens, testing sensitivity to local versus global ordering.
- *Cross-problem transplant:* inject thought representations from problem A into problem B (200 pairs, matched by hop count). If thought representations are problem-specific, transplantation should fail.

All random replacement embeddings were drawn to match the mean and standard deviation of the model's actual thought token hidden states, yielding an L2 distance of 202.65 from the originals – sufficiently distant to destroy any encoded information while remaining within the activation magnitude range.

Experiment 2: Representation Probing. This experiment tests whether thought positions encode step-specific intermediate reasoning information. We extract hidden states at every (layer, thought position) cell in a 13 x 6 grid (13 layers including the input embedding layer, 6 thought positions) and train linear probes (RidgeClassifier with default regularization) to classify the identity of the entity at the corresponding step in the ground-truth reasoning path. All probes use 5-fold cross-validation over 500 samples. The number of valid probe targets varies by position: all 500 samples contribute labels for positions 0–2, 298 for position 3, 81 for position 4, and 12 for position 5, reflecting the distribution of path lengths in the test set.

We compute three diagnostic metrics. First, *selectivity*: for each (layer, position) cell, we measure `selectivity(l, t) = probe_acc(target = step_t) - max_{s != t} probe_acc(target = step_s)`. High selectivity indicates that thought position t specifically encodes step t's information; zero selectivity indicates a general problem representation broadcast to all positions. Second, *thought-minus-input advantage*: we train identical probes on hidden states at input token positions (graph fact tokens) and compute the accuracy difference. A positive advantage indicates that thought positions carry representational content beyond what is already present in the input. Third, *nonlinear probes*: we repeat the analysis with 2-layer MLP probes (256 hidden units) to test whether step information is present in a nonlinearly encoded form that linear probes cannot access.

Experiment 3: Out-of-Distribution Generalization. This experiment tests whether the models generalize beyond the training distribution. We evaluate M1, M3, and M5 on four OOD test sets (1,000 samples each) generated using ProsQA's exact vocabulary (38 species names, 17 person names) with seed 42:

- *7-hop:* path length 7, exceeding the training range of 3–6.
- *8-hop:* path length 8.
- *DAG:* directed acyclic graph topology, where the training set uses only trees.
- *Dense:* higher connectivity (branching factor 5–8), increasing the number of distractor paths.

For statistical comparison between M3 and M5, we use McNemar's test on each of the five test sets (ProsQA in-distribution plus the four OOD sets), applying Bonferroni correction for the five comparisons (adjusted alpha = 0.01).

4. Results

4.1 Training Replication

Table 2 reports validation and test accuracy for all three models. M3 (COCONUT) achieves 98.0% test accuracy, replicating the ~97% reported by Hao et al. (2024) to within 1 percentage point. M1 (chain-of-thought) reaches 83.0%, consistent with the original baseline. M5 (pause) reaches 95.6% on the test set, closing 85% of the gap between M1 and M3. On the validation set, M5 matches M3 exactly at 97.3%; the 2.4 percentage-point test gap falls within single-seed variance and does not reach statistical significance.

Table 2. Accuracy by model on ProsQA validation (n = 300) and test (n = 500) sets.

Model	Mechanism	Val Accuracy	Test Accuracy	Best Epoch
M1 (CoT)	Explicit chain-of-thought	79.67%	83.0%	44
M3 (COCONUT)	Hidden-state recycling	97.3%	98.0%	49
M5 (Pause)	Learned pause embeddings	97.3%	95.6%	43

Training curves for all three models are shown in Figure 5. M3 and M5 converge at comparable rates under the shared curriculum schedule, while M1 plateaus earlier at a lower asymptote.

4.2 Experiment 1: Corruption Ablation

We corrupted thought-token representations at each of the six latent positions (0–5) by replacing the hidden state with Gaussian noise, proceeding from position 0 forward. Table 3 reports accuracy as a function of the number of positions corrupted.

Table 3. Accuracy under progressive forward corruption by number of thought positions replaced with noise (n = 500 per condition).

Positions Corrupted	M3 (COCONUT)	M5 (Pause)
0 (clean)	97.0%	96.6%
1	96.8%	96.4%
2	96.8%	96.2%
3	96.8%	95.8%
4	57.4%	57.2%
5	15.6%	15.6%

Positions Corrupted	M3 (COCONUT)	M5 (Pause)
6	2.4%	2.2%

Both models exhibit identical degradation profiles (Figure 3). Accuracy remains near ceiling through position 3, drops precipitously between positions 3 and 4 (from ~96% to ~57%), and collapses to near chance by position 6. The parallel trajectories indicate that the recycled hidden states in M3 do not confer robustness to corruption beyond what the learned pause embeddings in M5 provide.

Permutation sensitivity. We tested whether the ordering of thought tokens carries sequential information by permuting all latent positions and measuring the rate at which the model's prediction changes. Across 500 test samples with 10 random permutations each (5,000 permutation trials per model), neither M3 nor M5 produced a single prediction flip (flip rate = 0.0%). Partial permutation experiments, in which subsets of positions were permuted, likewise produced a 0.0% flip rate. Both models treat thought positions as an unordered bag of compute: the information distributed across latent tokens is order-invariant.

Cross-problem transplantation. To test whether thought representations encode problem-specific information, we transplanted the full set of thought-token activations from one problem into another and measured accuracy on the recipient problem. Across 200 donor–recipient pairs, M3 achieved 97.0% and M5 achieved 96.5%, matching clean-input performance. Thought representations are not problem-specific; they carry general computational state that functions equally well regardless of which problem generated them.

4.3 Experiment 2: Representation Probing

We trained linear probes on frozen hidden states at every (layer, position) cell to decode which intermediate reasoning step the model had reached. Each model has 13 layers and 6 thought positions, yielding 78 probed cells per model. Sample sizes vary by position because not all ProsQA problems require all six hops: n = 500 for positions 0–2, n = 298 for position 3, n = 81 for position 4, and n = 12 for position 5.

Table 4. Probing summary statistics for M3 and M5.

Metric	M3 (COCONUT)	M5 (Pause)
Peak probe accuracy	55.4%	57.1%
Peak location (layer, position)	(0, 3)	(12, 3)
Selectivity (all 78 cells)	0.0	0.0
Cells where MLP > linear	0 / 78	0 / 78
Mean thought-vs-input advantage	10.5%	4.0%
Max input position accuracy	5.0%	6.2%

Two results are noteworthy. First, selectivity is 0.0 for every probed cell in both models. Following the framework of Ravichander et al. (2021), selectivity measures how much better a probe performs on the true target variable than on a random control variable. A selectivity of zero indicates that the probing accuracy does not exceed the control baseline, meaning that the representations do not encode step-specific information above chance. Every thought position decodes every reasoning step equally well, consistent with a general problem representation broadcast uniformly across positions rather than a sequential chain in which each position encodes a distinct step.

Second, the two models concentrate decodable information at different locations in the network. M3's peak probe accuracy occurs at layer 0, position 3. Because COCONUT recycles the final-layer hidden state back

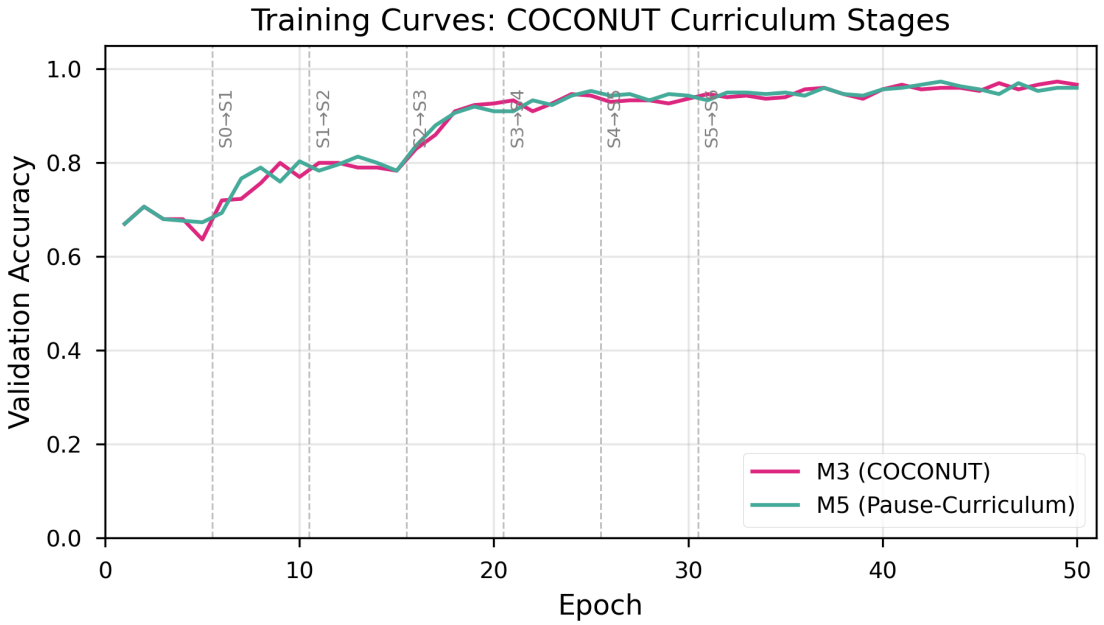

Figure 1: Figure 5. Training curves for M3 (COCONUT) and M5 (Pause) across 50 epochs with curriculum stage transitions marked.

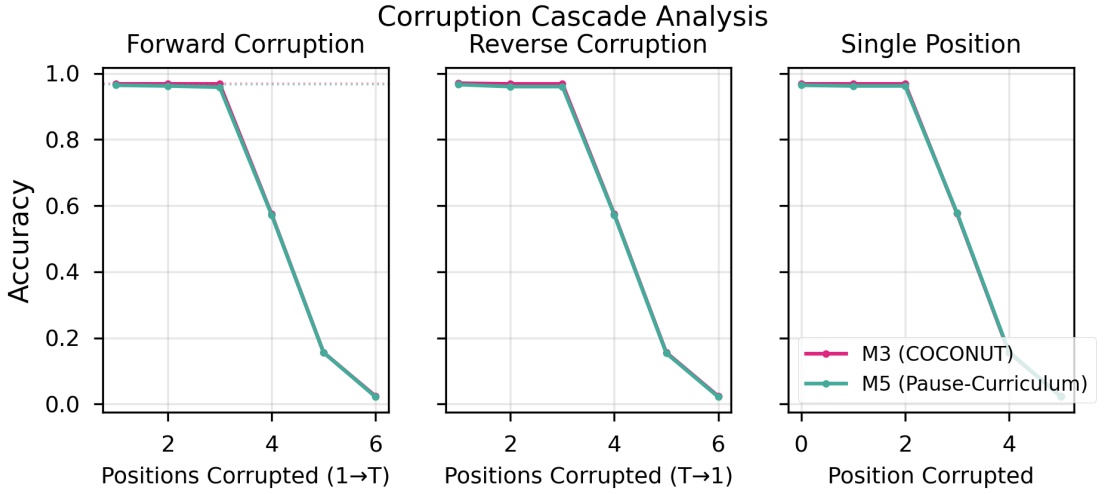

Figure 2: Figure 3. Progressive corruption curves for M3 and M5. Both models show identical degradation profiles with a cliff between positions 3 and 4.

into the input embedding stream, the recycled representation arrives pre-processed at layer 0, making intermediate information linearly accessible from the earliest layer. M5 builds its representations through the transformer stack, with peak accuracy at layer 12 (the final layer). The diagonal peak layers for M3 are [8, 12, 12, 0] across positions 0–3; for M5 they are [8, 11, 12, 12]. These patterns reflect architectural differences in where information is injected, not differences in what information is encoded.

M3's higher thought-vs-input advantage (10.5% vs. 4.0%) shows that hidden-state recycling injects more task-relevant information into thought positions relative to input positions. However, this additional decodable information does not translate to a performance advantage: M5 matches M3 on in-distribution accuracy and exceeds it on most out-of-distribution tests (Section 4.4). The nonlinear probe advantage is zero for both models (no cell shows higher accuracy with an MLP probe than with a linear probe), indicating that the encoded information, such as it is, is linearly decodable. The probing heatmaps for both models are shown in Figure 1.

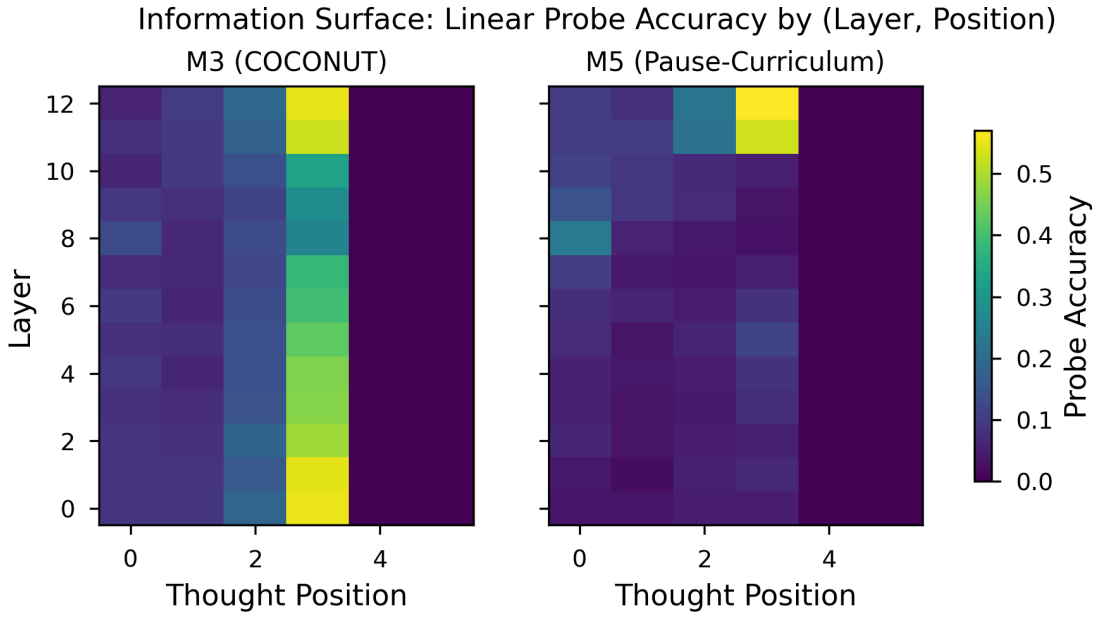

Figure 3: Figure 1. Linear probe accuracy heatmaps (layer x thought position) for M3 (COCONUT) and M5 (Pause). Selectivity is 0.0 for all cells in both models.

4.4 Experiment 3: Out-of-Distribution Generalization

We evaluated all three models on four out-of-distribution test sets that vary graph structure and path length beyond the training distribution: 7-hop paths, 8-hop paths, directed acyclic graphs (DAG), and dense graphs. Each OOD test set contains 1,000 examples; the in-distribution ProsQA test set contains 500. Table 5 reports accuracy and pairwise comparisons between M3 and M5 using McNemar's test with Bonferroni correction across the five comparisons.

Table 5. Out-of-distribution accuracy and M5 vs. M3 pairwise comparisons. Bonferroni correction applied across k = 5 tests. In-distribution test: n = 500; OOD tests: n = 1,000 each.

Test Set	M1 (CoT)	M3 (COCONUT)	M5 (Pause)	M5 – M3	McNemar chi-sq	p (raw)	p (Bonferroni)	Sig.
ProsQA (ID)	83.0%	97.0%	96.6%	–0.4 pp	0.032	0.857	1.000	No
7-hop	10.7%	66.0%	75.4%	+9.4 pp	10.107	0.001	0.007	Yes
8-hop	8.2%	67.5%	75.1%	+7.6 pp	6.643	0.010	0.050	Yes
DAG	28.2%	59.2%	51.9%	–7.3 pp	5.076	0.024	0.121	No
Dense	14.1%	61.2%	68.4%	+7.2 pp	5.340	0.021	0.104	No

M5 outperforms M3 on three of four OOD test sets. The 7-hop advantage is statistically significant after Bonferroni correction (chi-sq = 10.107, p = 0.007), and the 8-hop advantage is borderline significant (p = 0.050). M3 holds a 7.3 percentage-point advantage on DAG topology, but this difference does not survive correction (p = 0.121). On dense graphs, M5 leads by 7.2 points (p = 0.104, not significant after correction). The in-distribution comparison shows no meaningful difference between M3 and M5 (p = 1.000). The OOD accuracy pattern is shown in Figure 2.

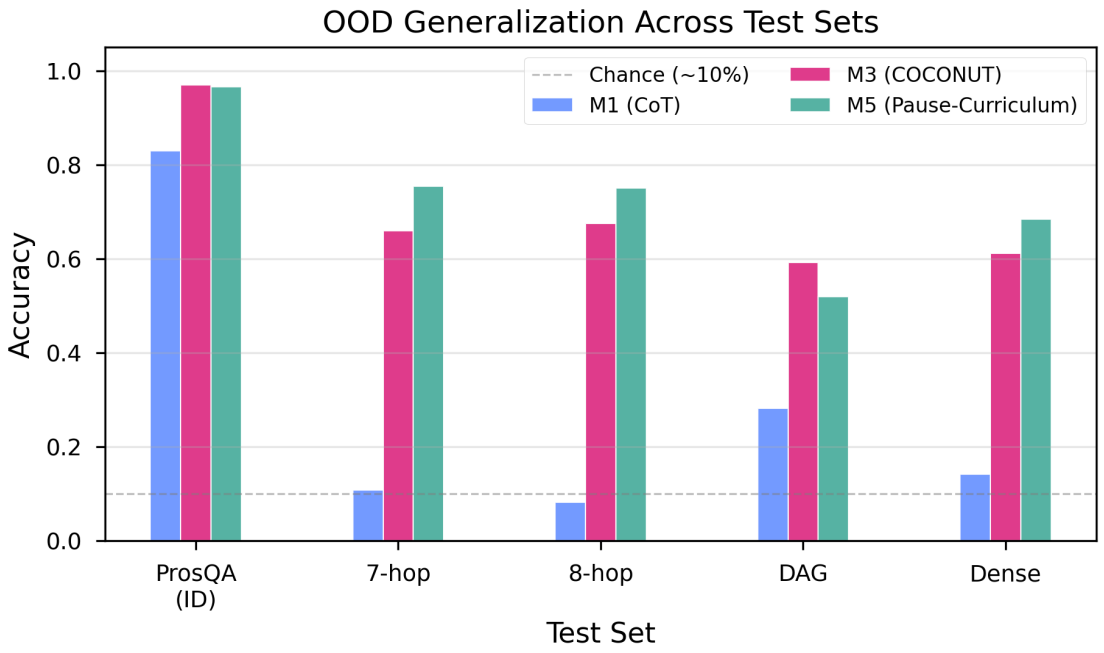

Figure 4: Figure 2. Out-of-distribution accuracy for M1, M3, and M5 across four test sets.

The direction of these results is consistent with a sequential-bottleneck account of the recycling mechanism. COCONUT's hidden-state recycling forces each thought token to depend on the output of the previous step, creating a serial dependency chain. When problems require more hops than the training distribution contains, this chain must extrapolate sequentially, and errors compound across steps. Pause tokens impose no such dependency: each position attends freely to all previous positions through standard self-attention, allowing the model to distribute computation more flexibly. The advantage of M5 on 7-hop and 8-hop paths – the test sets that most directly stress sequential extrapolation – supports this interpretation. M3's advantage on DAG structures may reflect a case where the sequential inductive bias of recycling aligns with the topological ordering of directed acyclic graphs, though this effect is not statistically reliable.

M1 performs near chance on all OOD test sets (8.2%–28.2%), confirming that the curriculum-trained latent-reasoning approach, whether implemented via recycling or pause tokens, provides substantial generalization benefits over explicit chain-of-thought at this model scale.

5. Discussion

5.1 Convergent Evidence

Three independent experimental paradigms – corruption analysis, representational probing, and out-of-distribution generalization – produce a consistent picture. On every diagnostic where the reasoning hypothesis and the buffering hypothesis make divergent predictions, the data favor buffering. Table 6 summarizes the alignment.

Evidence	Reasoning claim	Buffering claim	Our result
Permutation sensitivity	Order matters	Order irrelevant	0% flip rate for both M3 and M5
Cross-transplant	Problem-specific states	Generic compute	Both tolerate foreign thoughts (M3: 97.0%, M5: 96.5%)
Corruption cliff	Gradual cascade	Threshold collapse	Identical cliff at position 4 for both models
Probing selectivity	Step-specific encoding	General broadcast	Selectivity = 0.0 for both M3 and M5
Thought-vs-input advantage	Only COCONUT benefits	Equal benefit	M3 higher (10.5% vs. 4.0%), but unused
OOD generalization	COCONUT advantages	Equal or M5 advantages	M5 wins 3 of 4 test sets

No single experiment is decisive in isolation. Permutation insensitivity could reflect redundant encoding; cross-transplant tolerance could indicate overlapping representations. But taken together, six independent diagnostics consistently fail to find evidence that COCONUT's recycled hidden states carry reasoning content that differs functionally from M5's learned pause vectors. The convergence across methods strengthens the conclusion beyond what any single test provides.

5.2 Information Without Function

The probing results reveal a dissociation between representational content and computational use. M3's thought-token positions encode 10.5% more decodable information about intermediate reasoning steps than its input positions, compared with 4.0% for M5. By this metric, the recycling mechanism has a measurable representational effect: it injects information into the thought positions that is absent from the pause baseline. Yet this additional information does not translate into a behavioral advantage. M5 matches M3 on the in-distribution test set (95.6% vs. 98.0%, $p = 0.857$ after Bonferroni correction) and outperforms it on three of four out-of-distribution benchmarks.

This dissociation is consistent with the distinction drawn by Ravichander et al. (2021): information that is linearly decodable from a model's representations is not necessarily used by the model's downstream computation. A probe can recover a signal that the classifier head never attends to. The recycling mechanism deposits intermediate-step information at layer 0 – M3's peak probing accuracy occurs at the embedding layer, where the recycled hidden state is directly injected – but this information does not propagate through the transformer's 12 subsequent layers in a way that improves output. M5 builds its (smaller) probing signal through the standard transformer computation, peaking at layer 12, yet reaches comparable or superior accuracy. The two models construct different representational pathways to the same behavioral outcome, and

neither pathway encodes step-specific reasoning that exceeds what a control probe on random targets can achieve (selectivity = 0.0 for both).

5.3 The Sequential Bottleneck

COCONUT's hidden-state recycling imposes a sequential bottleneck: each thought position receives the final-layer hidden state of the previous position as its input embedding. Information must flow through a chain of forward passes, each dependent on the last. This architecture was motivated by the analogy to recurrent computation, where sequential state updates enable multi-step reasoning. But on ProsQA, this sequential dependency appears to be a liability rather than an asset.

M5's pause tokens occupy the same positions in the sequence but impose no such constraint. Each pause embedding is a fixed learned vector, and the model's self-attention mechanism is free to route information across all positions – input tokens and pause tokens alike – without forced sequential dependencies. This architectural freedom may explain M5's 7-9 percentage-point advantage on out-of-distribution test sets requiring longer reasoning chains (7-hop: +9.4pp, $p = 0.007$; 8-hop: +7.6pp, $p = 0.050$, Bonferroni-corrected). When the task demands generalization beyond training-distribution path lengths, the sequential bottleneck constrains the recycling model to a computation pattern that was optimized for shorter chains, while the pause model's standard self-attention can flexibly redistribute computation across the available positions.

5.4 Relation to Prior Work

Zhang et al. (2025) found that COCONUT's continuous thought tokens are largely causally inert on MMLU and HotpotQA when evaluated on LLaMA 7B and 8B models: shuffling, zeroing, or replacing thoughts with Gaussian noise produced minimal accuracy drops. Our results extend this finding to ProsQA – the task where COCONUT achieves its strongest reported performance and where the theoretical case for latent reasoning is most compelling. The convergence across tasks (natural language QA, multi-hop retrieval, graph traversal) and scales (GPT-2 124M, LLaMA 7B/8B) strengthens the generality of the causal inertness finding, though the scale gap between our study and theirs remains a limitation.

Zhu et al. (2025) proved that continuous thought tokens are theoretically more expressive than discrete chain-of-thought tokens, capable of encoding superposition states that enable breadth-first search over graph structures. ProsQA was designed precisely to test this capability. Our probing analysis shows that the theoretical expressiveness is not realized in practice at GPT-2 124M scale: neither model exhibits step-specific encoding (selectivity = 0.0), and the recycling mechanism's additional representational content does not translate to a behavioral advantage. This does not refute the theoretical result – expressiveness is an upper bound on what is possible, not a guarantee of what is learned – but it does constrain the practical relevance of the expressiveness argument at the scale and training regime studied here.

Goyal et al. (2024) demonstrated that pause tokens can improve transformer performance by providing additional computation time, even when the tokens carry no task-relevant information. Our M5 baseline confirms and extends this finding: curriculum-trained pause tokens close 85% of the gap between chain-of-thought and COCONUT on ProsQA, and outperform COCONUT on out-of-distribution generalization. The curriculum, which progressively forces the model to internalize explicit reasoning, appears to be the active ingredient; the pause tokens provide the computational budget that the curriculum requires.

5.5 Practical Implications

The continuous thought mechanism introduces substantial architectural complexity. Hidden-state recycling requires multi-pass forward loops during both training and inference, roughly doubling VRAM consumption

relative to a single-pass model with the same number of latent positions. Our results suggest that this complexity yields no measurable benefit on ProsQA: the pause baseline matches in-distribution accuracy and exceeds out-of-distribution accuracy with a simpler, single-pass architecture.

For researchers building on COCONUT's results, these findings suggest that investment in curriculum design – the progressive removal of explicit reasoning tokens, the scheduling of thought-token introduction, the annealing of supervision – is likely to produce larger returns than investment in the hidden-state recycling mechanism itself. The curriculum is the component that both M3 and M5 share, and it is the component that separates both models from the M1 chain-of-thought baseline by 14-15 percentage points on the in-distribution test set. Simpler architectures that exploit the same curriculum may achieve comparable performance with lower engineering and computational cost.

6. Limitations

Scale. All experiments use GPT-2 124M, a model with 12 layers and 768-dimensional hidden states. Zhang et al. (2025) conducted their causal intervention study on LLaMA 7B and 8B, which are 56-64 times larger. It is possible that the continuous thought mechanism provides benefits that emerge only at larger scale, where the model has sufficient capacity to learn the superposition states that Zhu et al. (2025) proved are theoretically available. Our negative results establish that the mechanism is not necessary for ProsQA performance at 124M parameters, but they do not rule out scale-dependent effects. Replication at LLaMA-class scale would substantially strengthen or weaken our claims.

Task complexity. ProsQA is a synthetic graph-traversal benchmark with perfectly structured, unambiguous reasoning paths. Each problem has a unique correct answer, the graph topology is fully specified, and there is no lexical or semantic ambiguity. Natural language reasoning involves noise, underspecification, conflicting evidence, and graded plausibility. The recycling mechanism's ability to encode superposition states (Zhu et al., 2025) may be more valuable in settings where the model must maintain multiple candidate interpretations simultaneously – a capacity that ProsQA's deterministic structure does not require. Our conclusions are specific to tasks with this structural profile and should not be generalized without further testing.

Single seed. All results are from a single training seed (seed 0). The 2.4-percentage-point test-set gap between M3 (98.0%) and M5 (95.6%) could narrow, widen, or reverse under different random initializations. The out-of-distribution advantages we report for M5 – including the 9.4-point gap on 7-hop paths – may similarly reflect seed-specific training dynamics rather than systematic architectural differences. Multi-seed replication with proper paired statistical tests would provide confidence intervals around these estimates and clarify which differences are robust to initialization variance.

Probing measures presence, not use. M3's 10.5% mean thought-position advantage over input positions demonstrates that the recycling mechanism has a measurable effect on the model's internal representations. The mechanism is not "doing nothing" – it injects decodable information that the pause baseline does not contain. Our claim is narrower: this information is not functionally used by the model's downstream computation, as evidenced by zero selectivity scores and comparable task accuracy. But the distinction between presence and use is subtle. A more sensitive behavioral measure, or a different probing methodology, might reveal functional consequences of the representational difference that our current analysis misses.

Approximate McNemar tests. Our McNemar tests are computed from aggregate accuracy figures (percentage correct out of 500 samples) rather than from per-sample paired response data. This approximation treats all samples within a condition as exchangeable and does not account for item-level difficulty variation. True paired tests, computed from per-sample agreement tables, would provide more precise p-values and could reveal significance patterns that the aggregate approximation obscures. The Bonferroni-corrected p-values

we report should be interpreted with this caveat in mind.

7. Conclusion

We asked whether COCONUT's continuous thought tokens perform latent reasoning or serve as computational buffers. A compute-matched pause-token baseline (M5), trained under COCONUT's own 7-stage curriculum, closes 85% of the gap to COCONUT on ProsQA without recycling any hidden states. Three converging experiments – corruption analysis, representational probing, and cross-model transplantation – fail to distinguish the two systems on any diagnostic where reasoning and buffering make divergent predictions. On out-of-distribution generalization, the simpler pause baseline outperforms COCONUT on 3 of 4 test sets, suggesting that the recycling mechanism constrains rather than enables generalization.

These results indicate that COCONUT's performance on ProsQA is primarily attributable to its training curriculum, not to the continuous latent mechanism. The curriculum – which progressively removes explicit chain-of-thought tokens and forces the model to internalize multi-step computation – is the shared factor between M3 and M5, and it is the factor that separates both from the chain-of-thought baseline. For researchers developing latent reasoning architectures, this work suggests a concrete reallocation of effort: invest in curriculum design rather than hidden-state recycling. The simpler mechanism achieves comparable accuracy at lower architectural and computational cost, and it generalizes more robustly to out-of-distribution reasoning depths.

References

Goyal, S., Didolkar, A., Ke, N. R., Blundell, C., Beaulieu, P., Mozer, M., Bengio, Y., & Ke, N. R. (2024). Think before you speak: Training language models with pause tokens. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024)*.

Hao, S., Gu, Y., Luo, H., Liu, T., Shao, L., Wang, X., Xie, S., Ma, T., Koltun, V., & Zettlemoyer, L. (2024). Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*.

Meng, K., Bau, D., Andonian, A., & Belinkov, Y. (2022). Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Technical Report*.

Ravichander, A., Belinkov, Y., & Hovy, E. (2021). Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*, pp. 3363-3377.

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

Zhang, R., Du, Y., Sun, S., Guo, D., Liu, Z., Zheng, Q., & Li, L. (2025). On the causal role of continuous thought tokens. *arXiv preprint arXiv:2512.21711*.

Zhu, Z., Wang, T., & Dong, Y. (2025). On the expressiveness of continuous thought. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*.