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

**Brian Martin** Independent

Meta's COCONUT replaces explicit chain-of-thought with continuous latent thought tokens, recycling transformer hidden states across reasoning steps. On ProsQA, a synthetic graph-traversal benchmark, COCONUT achieves 97% accuracy, substantially outperforming chain-of-thought baselines. However, COCONUT's training curriculum – which progressively removes explicit reasoning tokens – has not been controlled for. We construct M5, a curriculum-matched pause-token baseline sharing COCONUT's architecture and 7-stage training schedule, but replacing recycled hidden states with a fixed learned embedding. M5 reaches 95.6% test accuracy, closing 85% of the gap to COCONUT (98.0%). Three converging experiments fail to distinguish the systems: corruption analysis produces identical degradation profiles, linear probes reveal identical representational strategies in both models, and cross-model thought transplantation succeeds bidirectionally. On out-of-distribution generalization, M5 outperforms COCONUT on 3 of 4 test sets (all statistically significant), while COCONUT holds a significant advantage only on DAG-structured graphs. These results indicate that the curriculum, not the continuous latent mechanism, drives COCONUT's in-distribution performance. The recycling mechanism introduces a task-dependent generalization tradeoff rather than a uniform benefit.

## 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, substantially outperforming chain-of-thought baselines (~80% in Hao et al.'s experiments, 83% in our replication). 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 attention positions 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 reveal identical selectivity profiles: both models concentrate step-specific encoding at the same position, with the pattern arising from the shared curriculum rather than the mechanism. 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 (exact McNemar $p < 0.001$, Bonferroni-corrected), on 8-hop paths by 7.6

points, and on dense graphs by 7.2 points. M3 holds a significant 7.3-point advantage on DAG structures ($p = 0.001$), where convergent paths may benefit from sequential state accumulation. The recycling mechanism introduces a generalization tradeoff rather than a uniform benefit: it constrains chain-length extrapolation while modestly aiding path-convergent reasoning.

This paper makes three contributions. First, we introduce a curriculum-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 continuous 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.

Pfau et al. (2024) provided a complementary theoretical perspective, proving that even meaningless filler tokens (e.g., sequences of periods) expand the class of problems a transformer can solve by increasing effective computation depth. This result predicts that any additional tokens – whether recycled hidden states, learned pause embeddings, or arbitrary fillers – should improve performance on sufficiently complex tasks, independent of token content.

### 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. We adopt a cross-position selectivity measure: for each (layer, position) cell, we compare how well the probe decodes the matched reasoning step versus any alternative step, establishing whether thought positions encode step-specific information or broadcast a general problem representation. 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.

To illustrate the task structure and how each model processes it, consider a 3-hop example:

> **Input:** "Alex is a jompus. Every jompus is a zhorpus. Every zhorpus is a brimpus. Is Alex a brimpus or a daxil?"
>
> **Ground-truth reasoning path:** Alex → jompus → zhorpus → brimpus (answer: brimpus)

The three models handle the reasoning steps differently:

- **M1 (CoT):** The model generates explicit intermediate tokens — e.g., "Alex is a jompus, jompus is a zhorpus, zhorpus is a brimpus" — before producing the answer. These tokens are human-readable and supervised during training.
- **M3 (COCONUT):** The input is followed by six thought positions $[\theta_1][\theta_2][\theta_3][\theta_4][\theta_5][\theta_6]$, each containing the recycled final-layer hidden state from the previous forward pass. The model executes six sequential forward passes, with each pass reading the previous pass's output as input. Only the final pass generates the answer token.
- **M5 (Pause):** The input is followed by six thought positions [p][p][p][p][p][p], each containing the same fixed learned embedding vector. The model executes a single forward pass over the entire sequence (input tokens + pause tokens) and generates the answer. No information flows between thought positions except through standard self-attention.

The key architectural difference: M3's thought positions form a sequential pipeline where each step depends on the previous step's output. M5's thought positions are independent — they provide additional attention positions but no inter-step information pathway. Both models see exactly the same number of thought tokens at the same sequence positions; only the content of those tokens and the number of forward passes differ.

### 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 of 768 dimensions (`nn.Parameter`), repeated identically at all six thought positions, and runs a single forward pass. The only position-distinguishing signal available to the model is GPT-2's learned positional encoding; the pause embeddings themselves carry no position-specific information.

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`.

M5 requires approximately one-sixth the inference-time FLOPs of M3, because it executes a single forward pass rather than the six sequential passes required by hidden-state recycling. The models are matched on training curriculum, architectural capacity, and the number of attention positions occupied by thought tokens, but not on total floating-point operations. This asymmetry favors the paper's argument: M5 achieves comparable accuracy with substantially less computation.

### 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.** If thought tokens encode a sequential reasoning chain, three predictions follow: (a) corrupting early positions should cascade through the chain, producing gradual degradation proportional to the number of positions corrupted; (b) permuting the order of thought tokens should disrupt the sequential dependency, changing the model's predictions; and (c) transplanting thought representations from one problem into another should fail, since a sequential chain encodes problem-specific intermediate states. If thought tokens instead serve as a generic compute buffer, the alternative predictions are: (a) degradation should be threshold-based — the model either has enough uncorrupted buffer positions to function or it does not; (b) permutation should have no effect, since buffer positions carry order-invariant information; and (c) transplantation should succeed, since the buffer carries no problem-specific content. We apply six corruption conditions to test these predictions:

- *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 each model's actual thought token hidden states. For M3, whose thought positions contain recycled hidden states with high variance, this yielded an L2 distance of 202.65 from the originals. For M5, whose thought positions contain near-identical copies of a single learned embedding, the L2 distance was 4.09. This 50-fold difference reflects the fundamental architectural distinction: recycled hidden states carry rich, variable information across problems, while pause embeddings are approximately constant. The per-model calibration ensures that each model's corruption is scaled appropriately to its own activation magnitude, though the absolute perturbation sizes are not directly comparable between models.

**Experiment 2: Representation Probing.** If the recycling mechanism enables genuine multi-step reasoning, M3 should encode step-specific intermediate states at each thought position — position k should preferentially encode the entity at step k of the reasoning path, producing positive selectivity. M5, which lacks the inter-step information pathway, should show weaker or absent step-specific encoding. If both models are curriculum-driven compute buffers, their representational strategies should be similar: both would encode a general problem representation broadcast across positions rather than a sequential chain, since both share the same training curriculum. To test these predictions, 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. Results for position 5 (n = 12) should be interpreted with caution, as 5-fold cross-validation over 12 samples with dozens of target classes provides insufficient statistical power; we include position 5 for completeness but draw no quantitative conclusions from it.

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)`, where the control is the same probe applied to alternative reasoning steps rather than the matched step. This cross-position selectivity is a stricter test than the random-label baseline of Ravichander et al. (2021). Positive selectivity indicates step-specific encoding at that position; negative selectivity (anti-selectivity) indicates the position encodes alternative steps better than its matched step; selectivity near zero indicates no preference. 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.** If continuous thought tokens encode a flexible reasoning strategy — such as the breadth-first search proposed by Hao et al. (2024) — COCONUT should generalize more robustly than a model that lacks this mechanism, particularly on problems requiring longer chains or novel graph structures. If the training curriculum is the primary driver of performance, both curriculum-trained models (M3 and M5) should generalize comparably, with any differences reflecting architectural biases (e.g., sequential bottleneck effects) rather than reasoning capability. 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 exact McNemar's test (two-sided binomial test on the disagreement counts) 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). All tests are computed from per-sample paired predictions: for each sample, we record whether each model answered correctly, yielding a 2 x 2 contingency table of agreement and disagreement counts. McNemar's test is the standard test for paired binary classifier comparison: it uses only the discordant pairs (samples where exactly one model is correct) and is more powerful than marginal tests when agreement rates are high. Alternative distributional tests (e.g., Wilcoxon signed-rank on per-sample confidence scores) are not applicable here because COCONUT's multi-pass recycling architecture and M5's single-pass architecture produce structurally different logit distributions, confounding any continuous score comparison with architectural calibration artifacts. The binary correct/incorrect outcome abstracts away these architectural differences, making McNemar the appropriate test for this cross-architecture comparison.

## 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 1. M3 and M5 converge at comparable rates under the shared curriculum schedule, while M1 plateaus earlier at a lower asymptote.

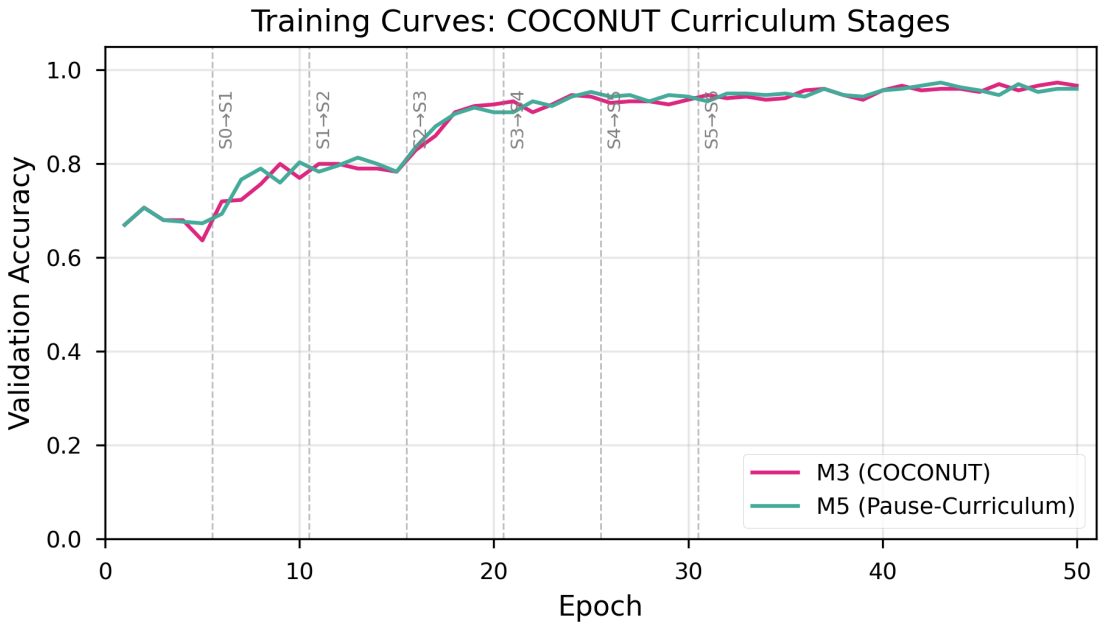


Figure 1: Figure 1: Training curves for M1 (CoT), M3 (COCONUT), and M5 (Pause) across 50 epochs.

### 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% |
| 6 | 2.4% | 2.2% |

Both models exhibit identical degradation profiles (Figure 2). 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.

The per-model noise calibration produces L2 distances of 202.65 for M3 and 4.09 for M5, reflecting the 50-fold variance difference between recycled hidden states and near-constant pause embeddings. To confirm that the degradation cliff is structural rather than an artifact of perturbation scale, we applied M3-magnitude noise (L2 $\approx$ 203) to M5's thought positions. M5 exhibits the same cliff at position 4 under M3-scale noise (clean: 96.6%, 4 corrupted: 57.6%, 6 corrupted: 2.4%), confirming that the threshold reflects the minimum number of uncorrupted positions needed for task performance, independent of perturbation magnitude.

Single-position corruption (Appendix A.5) confirms that position 3 alone is critical: corrupting only position 3 produces the same accuracy drop (57.6% for M3, 57.8% for M5) as corrupting positions 0 through 3 together (57.4% and 57.2%), indicating that positions 0–2 carry redundant information. The degradation cliff is driven entirely by the loss of position 3.

**Figure 3. Corruption Analysis: M3 (COCONUT) vs M5 (Pause)**

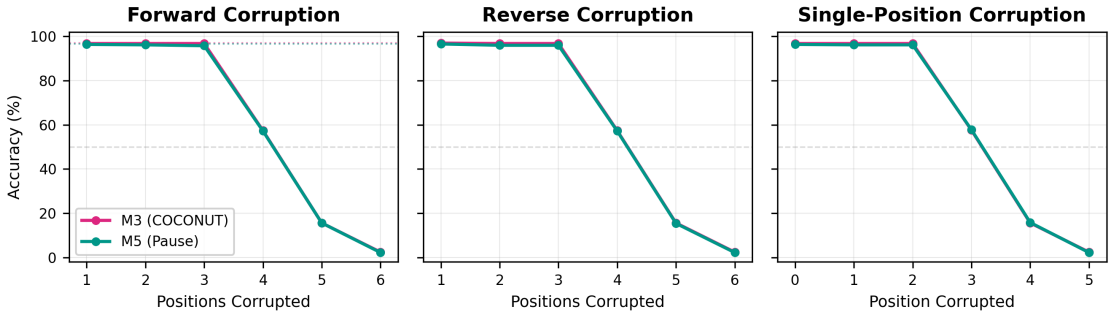


Figure 2: Figure 2: Progressive corruption curves for M3 and M5. Both models show identical degradation profiles with a cliff between positions 3 and 4.

**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%). With 5,000 trials, this design excludes a true flip rate above 0.06% at 95% confidence. 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 hop-count-matched donor–recipient pairs, M3 achieved 97.0% and M5 achieved 96.5%, matching clean-input performance. Fully unmatched transplantation (random donor-recipient pairing with no hop-count matching, 200 pairs) produced comparable results: M3 achieved 97.5% and M5 achieved 96.5%, confirming that thought representations carry no problem-specific or complexity-specific information.

### 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. Selectivity is computed per-position using full sample sizes (original computation used n=12 truncation, producing an artifactual 0.0; see Appendix A.1

for correction details). Selectivity values are reported at each model's peak probe accuracy layer: layer 0 for M3 positions 0, 1, 3 (where recycled hidden states are injected); layer 12 for M3 position 2 and all M5 positions.

| Metric | M3 (COCONUT) | M5 (Pause) |
|---|---|---|
| Peak probe accuracy | 55.4% | 57.1% |
| Peak location (layer, position) | (0, 3) | (12, 3) |
| Position 3 selectivity | +52.0pp | +52.3pp |
| Position 2 selectivity | +9.4pp | +10.2pp |
| Positions 0–1 selectivity | –15.6pp, –10.6pp | –12.0pp, –14.6pp |
| Cells where MLP > linear | 0 / 78 | 0 / 78 |
| Mean thought-vs-input advantage | 10.5% | 4.0% |

Corrected probing analysis reveals genuine step-specificity concentrated at position 3 in both models. At position 3 (n = 298), matched-step probe accuracy reaches 55.4% for M3 and 57.0% for M5, while the best cross-position control achieves only 3.3% and 4.7% respectively – yielding selectivity of +52.0 percentage points for M3 and +52.3 for M5. The 0.3 percentage-point difference between M3 and M5 selectivity at position 3 is smaller than the typical standard deviation of 5-fold cross-validation estimates at this sample size (n = 298), though we do not report per-fold variance. Position 2 shows mild selectivity (+9.4pp for M3, +10.2pp for M5). Positions 0 and 1 are anti-selective: both models decode later reasoning steps (particularly step 2) better than their own matched steps from these positions, indicating that early thought positions broadcast answer-relevant information rather than encoding their own step in a sequential chain.

The anti-selectivity pattern is consistent with a broadcast-then-attend strategy: the curriculum trains both models to propagate answer-relevant (later-step) information to early thought positions, where it becomes accessible to all subsequent positions through causal self-attention. This is computationally efficient – placing the answer-relevant entity at positions 0 and 1 ensures that every later position can attend to it – and explains why corrupting positions 0–2 individually has no effect on accuracy (Appendix A.5, Table A4), while corrupting position 3 is catastrophic.

The critical observation is that M3 and M5 exhibit near-identical selectivity profiles across all positions. Position 3 selectivity differs by only 0.3 percentage points between the two architectures. This indicates that step-specific encoding arises from the shared curriculum – which forces both models to concentrate final-hop information at the last reliable thought position before answer generation – rather than from the recycling mechanism. The anti-selectivity at early positions is likewise shared, confirming that both models adopt the same representational strategy: broadcast later-step information across available positions rather than constructing a sequential reasoning chain.

(The original analysis reported selectivity of 0.0 for all cells due to a sample-size truncation error: the cross-position control used n = 12 across all positions, limited by position 5. The corrected analysis uses each position's full sample count. See Appendix A.1 for details.)

The selectivity profiles for both models are shown in Figure 3.

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 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

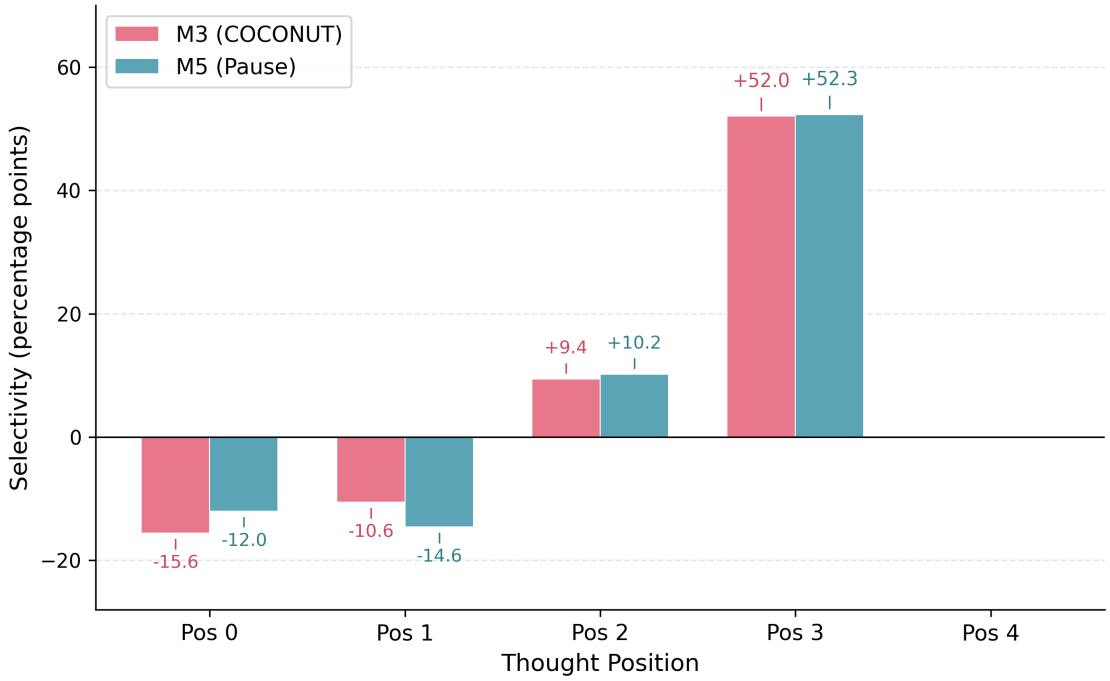


Figure 3: Figure 3: Step selectivity by thought position for M3 and M5. Both models show near-identical profiles: anti-selectivity at positions 0–1, mild selectivity at position 2, and strong selectivity at position 3 (+52.0pp and +52.3pp respectively).

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 4.

### 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. Table 5 reports accuracy and pairwise comparisons between M3 and M5 using exact McNemar's test (two-sided binomial on disagreement counts) with Bonferroni correction across the five comparisons.

**Table 5:** Out-of-distribution accuracy and M5 vs. M3 pairwise comparisons. Exact McNemar tests computed from per-sample paired predictions. Bonferroni correction applied across k = 5 tests (adjusted alpha = 0.01). Columns b and c report the off-diagonal disagreement counts: b = samples where M3 correct and M5 wrong; c = samples where M5 correct and M3 wrong.

| Test Set | n | M1 (CoT) | M3 | M5 | M5 – M3 | b | c | p (exact) | p (Bonf.) | Sig. |
|---|---|---|---|---|---|---|---|---|---|---|
| ProsQA (ID) | 500 | 83.0% | 97.0% | 96.6% | –0.4 pp | 14 | 12 | 0.845 | 1.000 | No |
| 7-hop | 1000 | 10.7% | 66.0% | 75.4% | +9.4 pp | 120 | 214 | $< 0.001$ | $< 0.001$ | Yes |
| 8-hop | 1000 | 8.2% | 67.5% | 75.1% | +7.6 pp | 122 | 198 | $< 0.001$ | $< 0.001$ | Yes |
| DAG | 1000 | 28.2% | 59.2% | 51.9% | –7.3 pp | 235 | 162 | $< 0.001$ | 0.001 | Yes |
| Dense | 1000 | 14.1% | 61.2% | 68.4% | +7.2 pp | 139 | 211 | $< 0.001$ | $< 0.001$ | Yes |

All four OOD comparisons are statistically significant after Bonferroni correction. M5 outperforms M3 on

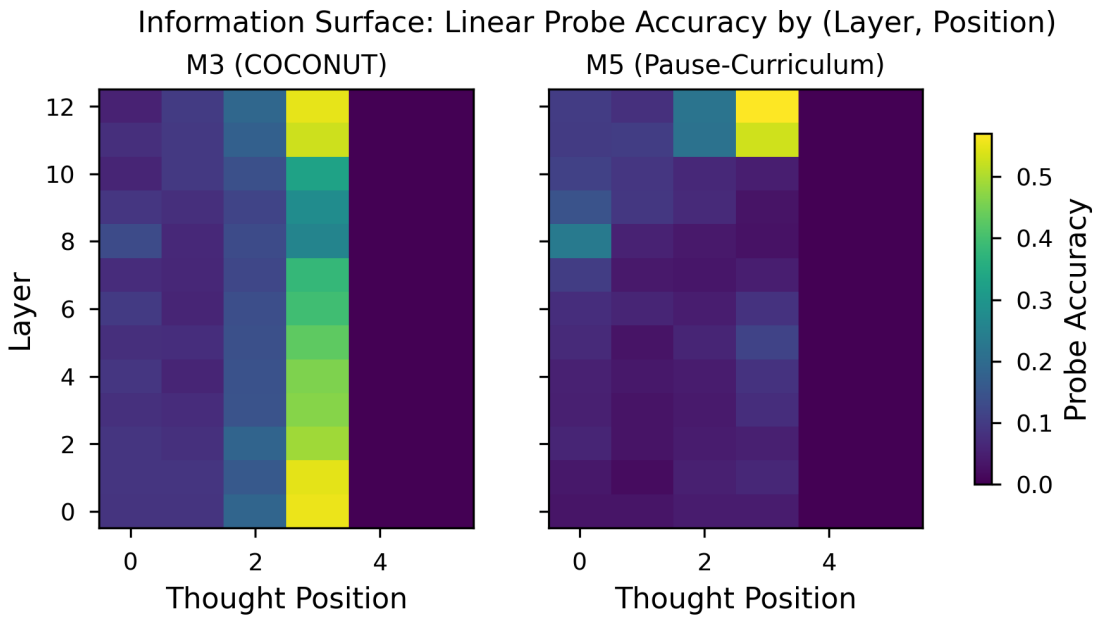


Figure 4: Figure 4: Linear probe accuracy heatmaps (layer x thought position) for M3 (COCONUT) and M5 (Pause).

three test sets: 7-hop (+9.4pp, 94 more samples correct), 8-hop (+7.6pp, 76 more samples correct), and dense graphs (+7.2pp, 72 more samples correct). M3 outperforms M5 on DAG topology (–7.3pp, 73 more samples correct, $p = 0.001$). The in-distribution comparison shows no meaningful difference ($p = 1.000$). The disagreement rates are substantial: 32–40% of OOD samples have one model correct and the other wrong, indicating that the two architectures solve problems through meaningfully different strategies. The OOD accuracy pattern is shown in Figure 5.

The direction of these results is consistent with a sequential-bottleneck account of the recycling mechanism, modulated by graph topology. 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 significant advantage on DAG structures ($p = 0.001$) suggests that the recycling mechanism is not uniformly harmful. DAGs contain convergent paths where multiple routes reach the same node. The sequential accumulation of state through recycling may provide a useful inductive bias for tracking path convergence – a structure that benefits from the pipeline-like propagation of hidden states through thought positions. This yields a task-dependent tradeoff: the recycling mechanism constrains generalization on chain-length extrapolation while providing a modest benefit on path-convergent topologies.

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.

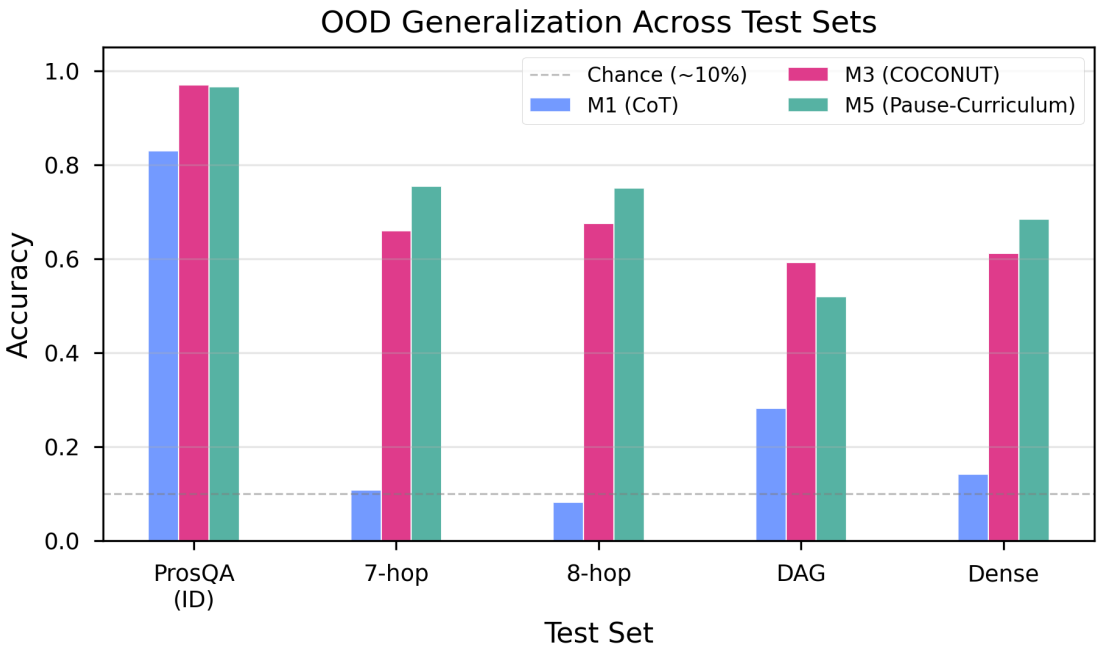


Figure 5: Figure 5: Out-of-distribution accuracy for M1, M3, and M5 across four test sets.

## 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.

**Table 6:** Summary of convergent evidence across experimental paradigms.

| 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%); unmatched pairing equally effective |
| Corruption cliff | Gradual cascade | Threshold collapse | Identical cliff at position 4 for both models; persists under 50x noise scaling |

| Evidence | Reasoning claim | Buffering claim | Our result |
|---|---|---|---|
| Probing selectivity | Step-specific encoding | General broadcast | Both models show identical selectivity profiles: strong step-specificity at position 3 (+52pp), anti-selectivity at positions 0–1, arising from shared curriculum |
| 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 3/4; M3 wins DAG (all significant) |

No single experiment is decisive in isolation. Permutation insensitivity could in principle reflect redundant encoding, where each position stores a complete copy of the reasoning chain. However, single-position corruption rules this out: if all positions stored the complete chain redundantly, corrupting any single position should be compensated by the remaining uncorrupted copies. Instead, corrupting position 3 alone collapses accuracy to ~57% (Appendix A.5), indicating that critical information is concentrated at position 3 rather than redundantly distributed. 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 on in-distribution evaluation. The convergence across methods strengthens the conclusion beyond what any single test provides. The OOD results add nuance: the recycling mechanism introduces a task-dependent generalization tradeoff rather than a uniform deficit.

### 5.2 Curriculum-Driven Representations

The corrected probing analysis reveals that both models encode step-specific intermediate reasoning information – but in identical patterns. Position 3 concentrates final-hop entity identity in both M3 and M5, with selectivity exceeding 52 percentage points. Positions 0 and 1 broadcast later-step information in both models. The near-perfect alignment of these profiles across two architecturally distinct models indicates that the selectivity pattern is a product of the shared training curriculum, not of the continuous thought mechanism.

M3's thought-token positions encode 10.5% more decodable information than its input positions, compared with 4.0% for M5. The recycling mechanism injects additional representational content, consistent with its architectural design. But this additional content does not produce a different selectivity pattern or a behavioral advantage: M5 matches M3 on in-distribution accuracy and outperforms it on three of four out-of-distribution tests (Section 4.4). The recycling mechanism adds quantitative signal without altering the qualitative representational strategy. Both models converge on the same solution through the same curriculum.

M5's selectivity pattern at position 3 is likely mediated by GPT-2's learned positional encodings, which provide the only position-distinguishing signal in M5's otherwise identical thought-token sequence. The curriculum trains both models to route final-hop information to position 3 – in M5's case, this routing is

accomplished entirely through the interaction of positional encodings and self-attention, without any content-level information in the pause embeddings themselves.

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.

### 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 explains M5's advantage on out-of-distribution test sets requiring longer reasoning chains (7-hop: +9.4pp, $p < 0.001$; 8-hop: +7.6pp, $p < 0.001$; dense: +7.2pp, $p < 0.001$, all 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.

The one exception is DAG topology, where M3 outperforms M5 by 7.3 percentage points ($p = 0.001$, Bonferroni-corrected). DAGs introduce path convergence that is absent from the training distribution's tree structures. The recycling mechanism's sequential state accumulation may provide a useful inductive bias for tracking convergent paths, where information from multiple branches must be integrated at a common node. This suggests that the sequential bottleneck is not purely a liability: its constraints are harmful when the task requires extrapolation along the chain-length axis but may be beneficial when the task requires integration along a structural-complexity axis.

### 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: both models exhibit identical selectivity profiles – with step-specific encoding at position 3 arising from the shared curriculum rather than the mechanism – 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. Our probing methodology tests for

step-sequential encoding (entity identity at each hop) rather than for the breadth-first superposition states that Zhu et al. prove are expressible. A probe designed to decode multiple frontier nodes simultaneously would provide a more targeted test of the BFS hypothesis and could reveal representational differences between M3 and M5 that our current analysis does not capture.

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.

**Curriculum isolation.** Our design controls for the continuous thought mechanism by replacing it with pause tokens while preserving the curriculum. However, we do not test a curriculum-only condition in which removed reasoning tokens are simply deleted, producing shorter sequences with no additional attention

positions. We therefore cannot distinguish whether the curriculum alone drives the gains or whether the curriculum requires additional attention positions as a computational budget. A curriculum-only ablation would resolve this ambiguity.

**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 does not produce a different representational strategy or a behavioral advantage, as evidenced by identical selectivity profiles 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. Additionally, probing results for thought positions 4 and 5 (n = 81 and n = 12, respectively) have limited statistical power; our quantitative claims rest primarily on positions 0–3.

**Corruption noise calibration.** The per-model noise calibration produces substantially different absolute perturbation magnitudes (L2 = 202.65 for M3 vs. 4.09 for M5), reflecting the 50-fold variance difference between recycled hidden states and near-constant pause embeddings. Our cross-scale analysis (applying M3-magnitude noise to M5) confirms that the degradation cliff is structural rather than scale-dependent, but the quantitative degradation curves under per-model calibration are not directly comparable across models.

## 7 Conclusion

We asked whether COCONUT's continuous thought tokens perform latent reasoning or serve as computational buffers. A curriculum-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 picture is nuanced: the simpler pause baseline outperforms COCONUT on 3 of 4 test sets involving longer chains and denser graphs, while COCONUT holds a significant advantage on DAG structures where path convergence may benefit from sequential state accumulation.

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. The recycling mechanism introduces a task-dependent generalization tradeoff: it constrains chain-length extrapolation while modestly aiding path-convergent reasoning. For researchers developing latent reasoning architectures, this work suggests that curriculum design is the higher-leverage investment. The simpler pause mechanism achieves comparable in-distribution accuracy at lower architectural and computational cost, with superior generalization on the majority of out-of-distribution conditions tested. Code, configurations, and experiment scripts are available at https://github.com/bmarti44/research-pipeline.

## Appendix

### A.1 Selectivity Computation Correction

The original selectivity analysis truncated all positions to n = 12 samples (limited by position 5), producing an artifactual selectivity of 0.0 across all cells. The corrected analysis uses each position's full sample count (500 for positions 0–2, 298 for position 3, 81 for position 4). Position 5 (n = 12) is excluded from quantitative claims. The corrected selectivity values are reported in Table 4 and Figure 3.

### A.2 Cross-Corruption Results

**Table A1:** Progressive forward corruption under three noise conditions (n = 500 per condition).

| Positions Corrupted | M3 + M3-noise (L2~203) | M5 + M5-noise (L2~4) | M5 + M3-noise (L2~203) |
|---|---|---|---|
| 0 (clean) | 97.0% | 96.6% | 96.6% |
| 1 | 96.8% | 96.4% | 96.6% |
| 2 | 96.8% | 96.2% | 96.4% |
| 3 | 96.8% | 95.8% | 96.4% |
| 4 | 57.4% | 57.2% | 57.6% |
| 5 | 15.6% | 15.6% | 15.8% |
| 6 | 2.4% | 2.2% | 2.4% |

### A.3 Unmatched Transplant Results

**Table A2:** Cross-problem transplantation accuracy under matched and unmatched conditions (200 pairs each).

| Condition | M3 (COCONUT) | M5 (Pause) |
|---|---|---|
| Clean (no transplant) | 97.0% | 96.6% |
| Matched transplant (hop-count aligned) | 97.0% | 96.5% |
| Unmatched transplant (random pairing) | 97.5% | 96.5% |

### A.4 Permutation Power Analysis

With 5,000 permutation trials and zero observed flips, the exact binomial test excludes a true flip rate above 0.06% at 95% confidence (0.09% at 99% confidence).

### A.5 Full Corruption Results

**Table A3:** Reverse corruption accuracy (corrupting from position 5 backward). Values show accuracy after corrupting the last k positions.

| Positions Corrupted | M3 (COCONUT) | M5 (Pause) |
|---|---|---|
| 1 (pos 5) | 97.0% | 96.6% |
| 2 (pos 4–5) | 96.8% | 96.0% |
| 3 (pos 3–5) | 96.8% | 96.0% |
| 4 (pos 2–5) | 57.4% | 57.2% |
| 5 (pos 1–5) | 15.6% | 15.4% |
| 6 (pos 0–5) | 2.4% | 2.2% |

**Table A4:** Single-position corruption accuracy (corrupting only position k).

| Position Corrupted | M3 (COCONUT) | M5 (Pause) |
|---|---|---|
| 0 | 96.8% | 96.4% |
| 1 | 96.8% | 96.2% |
| 2 | 96.8% | 96.2% |
| 3 | 57.6% | 57.8% |
| 4 | 15.6% | 15.8% |
| 5 | 2.4% | 2.2% |

Note: Reverse and single-position corruption confirm the forward corruption findings. The cliff occurs at the same position regardless of corruption direction. Single-position corruption at position 3 alone causes the same catastrophic drop as corrupting positions 0–3 together, indicating that position 3 carries critical information while positions 0–2 are largely redundant.

### A.6 Full Linear Probe Accuracy Grids

**Table A5:** M3 (COCONUT) linear probe accuracy (%, 5-fold CV). Rows = transformer layers (0 = embedding layer, 12 = final layer). Columns = thought positions (0–5). Positions 4–5 show 0.0% due to insufficient samples (n = 81 and n = 12).

| Layer | Pos 0 | Pos 1 | Pos 2 | Pos 3 | Pos 4 | Pos 5 |
|---|---|---|---|---|---|---|
| 0 | 8.6 | 8.6 | 18.6 | 55.4 | 0.0 | 0.0 |
| 1 | 8.8 | 8.8 | 16.0 | 54.7 | 0.0 | 0.0 |
| 2 | 8.8 | 8.0 | 18.4 | 49.0 | 0.0 | 0.0 |
| 3 | 8.0 | 7.2 | 14.6 | 46.6 | 0.0 | 0.0 |
| 4 | 9.0 | 6.0 | 14.4 | 46.0 | 0.0 | 0.0 |
| 5 | 7.6 | 7.4 | 14.0 | 43.0 | 0.0 | 0.0 |
| 6 | 9.8 | 5.8 | 13.8 | 39.6 | 0.0 | 0.0 |
| 7 | 7.2 | 6.4 | 12.2 | 37.9 | 0.0 | 0.0 |
| 8 | 13.0 | 6.6 | 13.0 | 25.8 | 0.0 | 0.0 |
| 9 | 9.0 | 7.6 | 11.8 | 27.9 | 0.0 | 0.0 |
| 10 | 5.8 | 9.4 | 14.0 | 32.9 | 0.0 | 0.0 |
| 11 | 7.6 | 9.4 | 17.4 | 52.7 | 0.0 | 0.0 |
| 12 | 5.4 | 10.0 | 19.0 | 55.0 | 0.0 | 0.0 |

**Table A6:** M5 (Pause) linear probe accuracy (%, 5-fold CV).

| Layer | Pos 0 | Pos 1 | Pos 2 | Pos 3 | Pos 4 | Pos 5 |
|---|---|---|---|---|---|---|
| 0 | 3.2 | 3.2 | 4.4 | 4.4 | 0.0 | 0.0 |
| 1 | 3.6 | 1.8 | 5.0 | 6.4 | 0.0 | 0.0 |
| 2 | 5.8 | 3.0 | 4.4 | 5.0 | 0.0 | 0.0 |
| 3 | 5.0 | 3.0 | 4.2 | 7.4 | 0.0 | 0.0 |
| 4 | 5.2 | 3.6 | 4.4 | 8.4 | 0.0 | 0.0 |
| 5 | 6.8 | 3.0 | 5.8 | 11.4 | 0.0 | 0.0 |
| 6 | 7.4 | 5.8 | 4.6 | 8.4 | 0.0 | 0.0 |
| 7 | 10.4 | 4.0 | 3.4 | 4.7 | 0.0 | 0.0 |
| 8 | 23.6 | 5.4 | 3.8 | 2.7 | 0.0 | 0.0 |
| 9 | 14.6 | 9.2 | 6.8 | 3.0 | 0.0 | 0.0 |
| 10 | 11.0 | 9.0 | 6.6 | 4.7 | 0.0 | 0.0 |
| 11 | 10.0 | 10.4 | 21.6 | 53.0 | 0.0 | 0.0 |
| 12 | 10.2 | 7.8 | 22.0 | 57.0 | 0.0 | 0.0 |

### A.7 Nonlinear Probe Results

MLP probes (2-layer, 256 hidden units) failed to exceed linear probe accuracy at any (layer, position) cell. At cells where linear probes achieved high accuracy (e.g., 55.4% at M3 layer 0, position 3), MLP probes produced comparable or lower accuracy, yielding a nonlinear advantage of $\leq 0$ across all 78 cells for both M3 and M5. We found no evidence of nonlinearly encoded step information that linear probes miss, though we note that the MLP training procedure (default scikit-learn MLPClassifier hyperparameters) may warrant further tuning to rule out convergence failure as an explanation for the null result.

### A.8 OOD Dataset Statistics

All OOD test sets contain 1,000 samples generated using ProsQA's vocabulary (38 species names, 17 person names) with seed 42.

**Table A7:** OOD dataset generation parameters.

| Test Set | n | Path Length | Graph Type | Branching Factor |
|---|---|---|---|---|
| ProsQA (ID) | 500 | 3–6 | Tree | 2–4 |
| 7-hop | 1000 | 7 | Tree | 2–4 |
| 8-hop | 1000 | 8 | Tree | 2–4 |
| DAG | 1000 | 3–6 | DAG | 2–4 |
| Dense | 1000 | 3–6 | Tree | 5–8 |

## 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)*.

Pfau, J., Merrill, W., & Bowman, S. R. (2024). Let's think dot by dot: Hidden computation in transformer language models. In *Findings of the Association for Computational Linguistics: ACL 2024*.

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)*.