The Curriculum Is the Mechanism: Dissecting COCONUT's Latent Thought Gains 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 two curriculum-matched controls: M3, a single-pass pause-token baseline that replaces recycled hidden states with a fixed learned embedding, and M4, a multi-pass control that matches COCONUT's sequential processing structure while using fixed embeddings, enabling factorial decomposition of the recycled-content and sequential-processing factors. M3 reaches 96.6% test accuracy, not significantly different from COCONUT's 97.0% (McNemar p = 0.845); M4 reaches 94.8% (p = 0.354 after Bonferroni correction). Three converging experiments probe the distinction: corruption analysis reveals identical degradation profiles for M2 and M3, linear probes reveal identical selectivity profiles despite COCONUT encoding information more broadly across layers (29/78 vs. 11/78 significant probing cells), and cross-model thought transplantation succeeds bidirectionally. On out-of-distribution generalization, factorial decomposition via M4 cleanly separates two confounded factors: recycled content actively hurts chain-length extrapolation (M4 outperforms M2 by 10.9pp on 7-hop, $p < 0.001$), while sequential processing drives DAG generalization (M4 outperforms M3 by 7.9pp, $p < 0.001$). Wilcoxon signed-rank tests on teacher-forced log-probabilities confirm that recycled content carries reasoning-relevant information (M2 assigns significantly higher confidence than M4, r = 0.678, p < 10^{-50}), but this higher confidence does not translate to better accuracy. At GPT-2 124M scale, the training curriculum drives COCONUT's accuracy on ProsQA; the continuous thought mechanism contributes measurably higher confidence (Wilcoxon r = 0.678 on in-distribution data) but does not improve accuracy.

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 two controls designed to resolve this confound. M3 is a single-pass pause-token baseline that shares every architectural and training detail with the COCONUT model (M2) — the same GPT-2 124M backbone, the same 7-stage curriculum schedule, and the same number of latent thought positions — but replaces the recycled hidden-state embeddings with fixed learned pause vectors (Goyal et al., 2024). M4 extends M3 by matching COCONUT's sequential multi-pass processing structure while retaining the fixed pause embeddings, creating a clean factorial design that separates the contribution of recycled content from sequential processing. If the curriculum is the primary driver, both M3 and M4 should match M2. If the recycling mechanism matters, only M4 (which eliminates recycling while preserving the processing structure) will reveal where the mechanism contributes.

M3 reaches 96.6% test accuracy, not significantly different from COCONUT's 97.0% (exact McNemar $p = 0.845$, $n = 500$, 95% CI for difference: [–2.4, +1.6] percentage points). M4 reaches 94.8% test accuracy; the 2.2 percentage-point gap from M2 does not reach significance after Bonferroni correction (exact McNemar $p = 0.071$, p_Bonferroni = 0.354). Three additional experiments converge on the same conclusion. Corrupting thought tokens produces identical degradation profiles for M2 and M3, with zero sensitivity to permutation order; linear probes trained on intermediate representations reveal identical selectivity profiles for M2 and M3: both models concentrate step-specific encoding at position 3, with the pattern arising from the shared curriculum rather than the mechanism. COCONUT does encode information more broadly — 29 of 78 probed cells show significant step decoding versus 11 for M3 — but this richer encoding produces no behavioral advantage. Cross-model thought transplantation succeeds bidirectionally, confirming that neither model's latent representations carry privileged information. On out-of-distribution test sets, M3 outperforms M2 on 7-hop, 8-hop, and dense graphs by 7–9 percentage points, while M2 holds a 7.3-point advantage on DAG structures. M4 enables a clean factorial decomposition of these OOD differences: the M4 vs. M2 comparison (matched processing, different content) reveals that recycled content harms chain-length generalization (M4 outperforms M2 on 7-hop by 10.9pp and 8-hop by 7.7pp, both $p < 0.001$ after Bonferroni correction), while the M4 vs. M3 comparison (matched content, different processing) shows that sequential processing drives DAG generalization (M4 outperforms M3 by 7.9pp, $p < 0.001$) but does not affect chain-length extrapolation (M4 and M3 do not differ on 7-hop or 8-hop).

This paper makes three contributions. First, we introduce a factorial control methodology — using both a single-pass and a multi-pass pause-token baseline — that isolates the curriculum from the mechanism and separately identifies the contributions of recycled content and sequential processing in latent-reasoning systems. This methodology is 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, using the factorial decomposition enabled by M4, we characterize the separate contributions of recycled content and sequential processing to out-of-distribution generalization, showing that recycled content harms chain-length extrapolation while sequential processing drives topological generalization.

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. Quiet-STaR (Zelikman et al., 2024) trains models to generate internal rationales at every token position and learn from downstream signal, internalizing reasoning without requiring explicit supervision. Deng et al. (2024) demonstrated that models can be distilled from explicit chain-of-thought into implicit reasoning through progressive removal of intermediate steps, suggesting that the training curriculum itself – rather than any particular latent mechanism – may be the key ingredient. COCONUT is the most prominent latent-reasoning architecture, 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 Extra Computation

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 M3 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 four 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.
- **M2 (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 position. The model first processes the input prefix in a single forward pass, then processes each thought token sequentially via KV-cache incremental decoding: the previous position's final-layer hidden

state becomes the current position's input embedding. Only after all thought positions are processed does the model generate the answer token.

- **M3 (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.
- **M4 (Pause-Multipass):** The input is followed by six thought positions, each containing the same fixed learned embedding vector — identical to M3. However, M4 processes these positions sequentially across 6 passes using KV-cache incremental decoding, matching M2's processing structure exactly. At each step, the same fixed embedding is re-injected; no hidden state is recycled. M4 isolates whether COCONUT's advantages come from the recycled content or from the sequential processing structure.

The key architectural relationships: M2 and M4 share the same sequential processing pipeline but differ in what is injected at each step (recycled hidden states vs. fixed embeddings). M3 and M4 share the same fixed embeddings but differ in processing structure (single pass vs. 6 sequential passes). All three curriculum-trained models see the same number of thought tokens at the same sequence positions.

3.2 Models

We train four 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. M1 = CoT baseline, M2 = COCONUT, M3 = Pause, M4 = Pause-Multipass. All share the same pretrained initialization, optimizer, and hyperparameters. M2, M3, and M4 share the same curriculum schedule.

Model	Thought mechanism	Processing	Curriculum
M1 (CoT)	None – explicit text reasoning tokens	Single pass	No stages (standard supervised)
M2 (COCONUT)	Hidden states from the previous forward pass recycled as input embeddings	6 sequential passes	7-stage progressive CoT removal
M3 (Pause)	Fixed learned pause embedding (`nn.Parameter`) at each thought position	Single pass	Same 7-stage curriculum as M2
M4 (Pause-Multipass)	Fixed learned pause embedding (`nn.Parameter`) at each thought position	6 sequential passes	Same 7-stage curriculum as M2

M3 is the primary 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: M2 recycles hidden states across multiple forward passes, creating a sequential information

pathway between thought steps, while M3 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.

However, M2 and M3 differ in two confounded ways: (1) the *content* of thought-token embeddings (recycled hidden states vs. fixed pause vectors) and (2) the *sequential processing structure* (6-pass incremental decoding vs. single-pass parallel). M4 (Pause-Multipass) resolves this confound. M4 uses the same fixed learned pause embedding as M3 but processes thought tokens sequentially across 6 passes, matching M2's processing structure exactly. At each step, the same fixed embedding is re-injected – no hidden state is recycled. This creates a clean factorial decomposition of the two confounded factors:

- **M2 vs. M4:** Same sequential processing, different content → isolates recycled content
- **M3 vs. M4:** Same fixed content, different processing → isolates sequential processing
- **M2 vs. M3:** Both factors differ → confounded (for reference only)

If M4 matches M2, the sequential processing structure drives any differences between M2 and M3, and the recycled content is inert. If M4 matches M3, the single-pass architecture is sufficient regardless of processing structure, and M2's advantages arise from recycled content. If M4 falls between M2 and M3, both factors contribute.

We implemented M3 and M4 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 (M2). When `feedback_mode="pause_curriculum"`, thought positions receive a learned `nn.Parameter` embedding and inference executes a single forward pass (M3). When `feedback_mode="pause_multipass"`, thought positions receive the same learned embedding but are processed sequentially across 6 passes, matching M2's KV-cache incremental decoding structure (M4). 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) new configuration files (`prosqa_m5_pause.yaml` for M3, `prosqa_m6_pause_multipass.yaml` for M4) identical to the COCONUT configuration except for the `feedback_mode` setting. No changes were made to `dataset.py` or `utils.py`.

M3 requires substantially fewer inference-time FLOPs than M2. COCONUT's recycling loop processes thought tokens sequentially: after a full forward pass over the input prefix, each subsequent thought token is processed as a single-token forward pass using KV-cache incremental decoding, with the previous position's final-layer hidden state injected as the current position's input embedding. M3 instead processes all thought tokens in a single forward pass alongside the input. M4 matches M2's sequential processing structure exactly – it processes thought tokens one at a time via KV-cache incremental decoding across 6 passes – but re-injects the same fixed pause embedding at each step rather than recycling content. The three models are thus matched on training curriculum, architectural capacity, and the number of attention positions occupied by thought tokens. M2 and M4 are additionally matched on sequential processing structure and total FLOPs, differing only in what is injected at each step. M3 and M4 differ only in whether thought tokens are processed sequentially or in parallel. This factorial design allows attribution of any performance differences to either the recycled content (M2 vs. M4) or the sequential processing structure (M3 vs. M4).

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 (M2, M3, and M4), 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 M2 and fixed pause embeddings for M3. 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; M2, M3, and M4 each required approximately 28–40 hours due to the multi-pass forward loop (M2, M4) and the longer sequences with thought tokens (all curriculum models). M4 is the slowest due to processing 6 sequential passes with fixed embeddings, which prevents the KV-cache optimization that COCONUT's recycled states enable.

3.4 Experiments

We design three experiments, each probing whether the continuous thought mechanism or the training curriculum drives COCONUT's performance. 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 M2, whose thought positions contain recycled hidden states with high variance, this yielded an L2 distance of 202.65 from the originals. For M3, 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, M2 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. M3, 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, using a grid search over 72 hyperparameter configurations (6 hidden sizes $\times$ 3 learning rates $\times$ 4 regularization strengths; see Appendix A.7), 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, all curriculum-trained models (M2, M3, and M4) should generalize

comparably, with any differences reflecting architectural biases rather than reasoning capability. The inclusion of M4 allows factorial attribution: if M2's OOD advantages over M3 arise from sequential processing rather than recycled content, M4 should match M2; if they arise from recycled content, M4 should match M3. We evaluate M1, M2, M3, and M4 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 comparisons, 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), applied to three pairwise comparisons: M2 vs. M3, M2 vs. M4, and M3 vs. M4. Bonferroni correction is applied within each comparison family. 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. The M2 vs. M4 comparison is the cleanest test of recycled content, as both models share the same sequential processing structure; the M3 vs. M4 comparison is the cleanest test of sequential processing, as both share fixed pause embeddings. The M2 vs. M3 comparison is retained for continuity with the two-model analysis but is now interpretable through the factorial decomposition.

4 Results

4.1 Training Replication

Table 2 reports validation and test accuracy for all four models. M2 (COCONUT) achieves 97.0% test accuracy, replicating the ~97% reported by Hao et al. (2024). M1 (chain-of-thought) reaches 83.0%, consistent with the original baseline. M3 (pause) reaches 96.6% on the test set, not significantly different from M2 (exact McNemar p = 0.845, 95% CI for accuracy difference: [–2.4, +1.6] percentage points). On the validation set, M3 matches M2 exactly at 97.3%. The 0.4 percentage-point test gap does not approach significance: only 26 of 500 samples are discordant (14 where M2 alone is correct, 12 where M3 alone is correct). M4 (pause-multipass) reaches 94.8% on the test set (96.7% validation, best epoch 30). The 2.2 percentage-point gap between M4 (94.8%) and M2 (97.0%) does not reach significance after Bonferroni correction (exact McNemar p = 0.071, p_Bonferroni = 0.354, 31 discordant pairs: 21 where M2 alone is correct, 10 where M4 alone is correct). M4 and M3 likewise do not differ significantly (–1.8pp, p_Bonferroni = 0.680).

Table 2: Accuracy by model on ProsQA validation (n = 300) and test (n = 500) sets. Test accuracy is reported from the independent experiment inference pipeline used throughout this paper. Training-time evaluation at best epoch yielded slightly higher estimates for M2 (98.0%) and lower for M3 (95.6%), a discrepancy of 5 samples per model attributable to differences in the inference code path; we use the experiment-pipeline numbers for consistency with all subsequent analyses.

Model	Mechanism	Processing	Val Accuracy	Test Accuracy	Best Epoch
M1 (CoT)	Explicit chain-of-thought	Single pass	79.67%	83.0%	44
M2 (COCONUT)	Hidden-state recycling	6 sequential passes	97.3%	97.0%	49
M3 (Pause)	Learned pause embeddings	Single pass	97.3%	96.6%	43
M4 (Pause-Multipass)	Learned pause embeddings	6 sequential passes	96.7%	94.8%	30

Training curves for all four models are shown in Figure 1. M2, M3, and M4 converge at comparable rates under the shared curriculum schedule, while M1 plateaus earlier at a lower asymptote. M4's 94.8% test accuracy falls 2.2pp below M2 ($p = 0.071$ uncorrected, $p = 0.354$ Bonferroni-corrected) — a gap that, while not significant, motivates examining the out-of-distribution comparisons where recycled content and sequential processing can be factorially decomposed (Section 4.4).

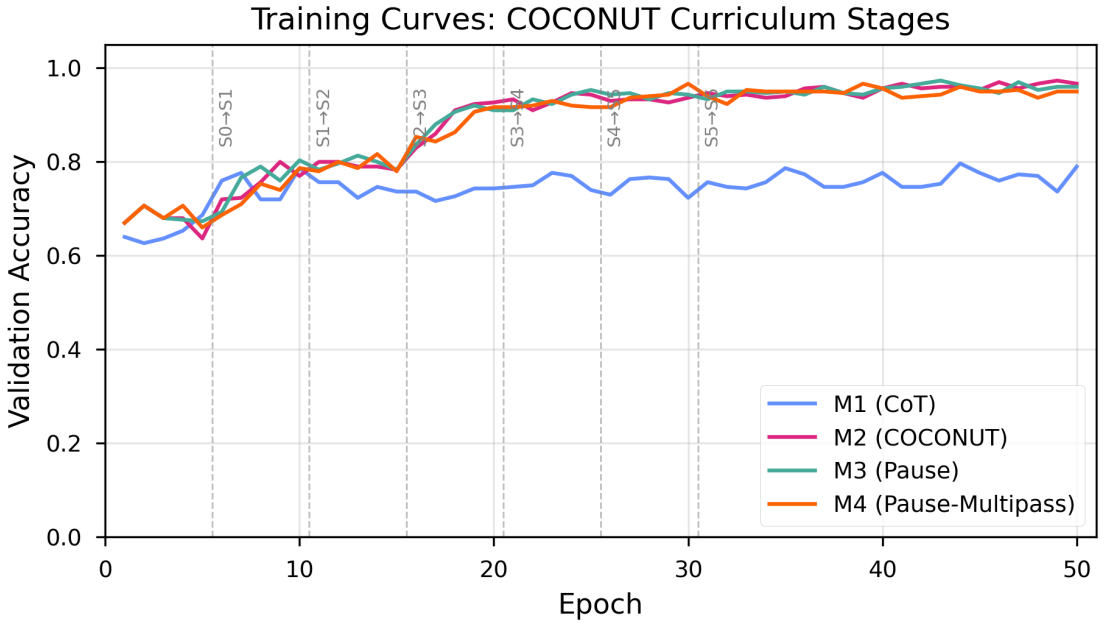

Figure 1: Figure 1: Training curves for M1 (CoT), M2 (COCONUT), M3 (Pause), and M4 (Pause-Multipass) across 50 epochs.

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	M2	M3
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%

Note: M4 is excluded from corruption analysis. The corruption methodology extracts hidden states via a single forward pass over the input embeddings, discarding the accumulated KV-cache state that defines M4's multi-pass computation. Corrupted representations injected into this artifact pipeline produce chance-level accuracy (2.4%) regardless of the number of positions corrupted, including the zero-corruption control, confirming that the methodology is incompatible with multi-pass KV-cache architectures rather than revealing genuine fragility. Extending corruption analysis to M4 would require per-pass embedding injection with KV-cache preservation, which we leave for future work.

M2 and M3 exhibit nearly identical degradation profiles (Figure 2). The maximum absolute accuracy difference between M2 and M3 at any corruption level is 1.0 percentage points (at 3 positions corrupted: 96.8% vs. 95.8%), within expected sampling variability for n = 500. 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 M2 do not confer robustness to corruption beyond what the learned pause embeddings in M3 provide.

The per-model noise calibration produces L2 distances of 202.65 for M2 and 4.09 for M3, 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 M2-magnitude noise (L2 $\approx$ 203) to M3's thought positions. M3 exhibits the same cliff at position 4 under M2-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 M2, 57.8% for M3) as corrupting positions 0 through 3 together (57.4% and 57.2%), indicating that positions 0–2 carry mutually redundant information – each encodes similar answer-relevant content (consistent with the anti-selectivity pattern in Section 4.3), so corrupting any individual early position is compensated by the remaining copies. The degradation cliff is driven entirely by the loss of position 3.

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 M2 nor M3 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

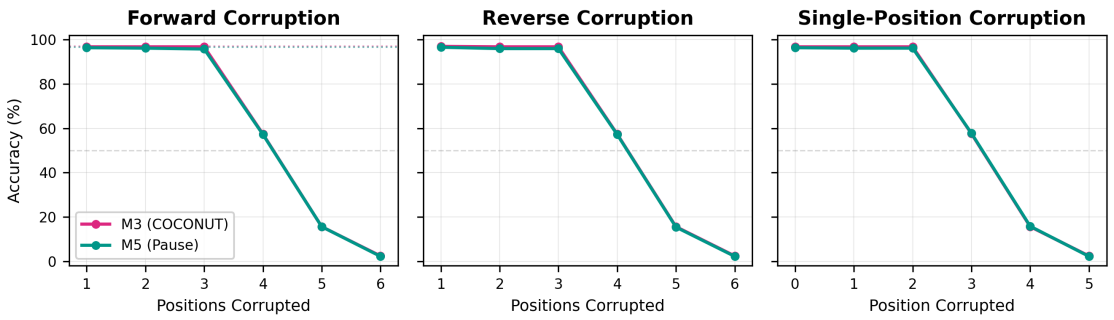

Figure 2: Figure 2: Progressive corruption curves for M2 and M3. Both models show identical degradation profiles with a cliff between positions 3 and 4.

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 with respect to final predictions: permuting latent tokens does not change the model's output. This does not rule out order-sensitive internal representations that are ultimately redundant for the final prediction.

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, M2 achieved 97.0% and M3 achieved 96.5%, matching clean-input performance. Fully unmatched transplantation (random donor-recipient pairing with no hop-count matching, 200 pairs) produced comparable results: M2 achieved 97.5% and M3 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 M2 and M3. Sample sizes vary by position: n = 500 (positions 0–2), n = 298 (position 3), n = 81 (position 4), n = 12 (position 5); absolute probe accuracies are not directly comparable across positions due to these differences. 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). For M2, selectivity is reported at layer 0 for positions 0, 1, and 3 (where recycled hidden states are injected) and layer 12 for position 2. For M3, selectivity is reported at layer 12 for all positions, matching the final-layer convention. At M3's peak accuracy layers (layer 8 for position 0, layer 11 for position 1), selectivity values differ: position 0 shows +17.0pp (positive, not anti-selective) at layer 8, while position 1 shows –11.2pp at layer 11; the anti-selectivity pattern at early positions is thus layer-dependent for M3. MLP probe results from grid search over 72 hyperparameter configurations (Appendix A.7). M4 probing results are not available; see note below.

Metric	M2	M3
Peak probe accuracy	55.4%	57.0%
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
Significant cells (Bonferroni)	29 / 78	11 / 78
MLP vs. linear at position 3	–9.4pp (MLP overfits)	–11.4pp (MLP overfits)
MLP vs. linear at position 2	**+10.2pp**	**+7.6pp**
Mean thought-vs-input advantage	10.5%	4.0%

M4 probing limitation. M4 probing results are omitted because the hidden-state extraction method is incompatible with the multi-pass architecture. The extraction code (`get_hidden_states`) performs a cold-start forward pass over the final embeddings without the KV-cache accumulated during the multi-pass loop. For M4, this captures only the representation of the fixed pause embedding processed in isolation, not the representation built through sequential KV-cache accumulation across 6 passes. Correcting this would require extracting hidden states from within the multi-pass loop at each pass, which is architecturally non-trivial and left to future work.

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 M2 and 57.0% for M3, while the best cross-position control achieves only 3.3% and 4.7% respectively – yielding selectivity of +52.0 percentage points for M2 and +52.3 for M3. The 0.3 percentage-point difference between M2 and M3 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 M2, +10.2pp for M3). 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 M2 and M3 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. The same truncation error invalidated the permutation-based significance tests in the original run, producing uniformly non-significant p-values (p = 1.0 for all 78 cells). We reran permutation tests with corrected sample sizes (2,000 permutations per cell, Bonferroni threshold = 0.05/78 = 0.000641). M2 yields 29/78 significant cells: all 13 layers at positions 2 and 3, plus scattered cells at positions 0–1. M3 yields 11/78 significant cells, concentrated in late layers at positions 0–2 and position 3. The differential — nearly 3× more significant cells for M2 — indicates that hidden-state recycling produces more broadly distributed and robustly decodable intermediate representations. However, this richer encoding does not translate to a behavioral advantage (Section 5.1). See Appendix A.1 for the full corrected permutation results.)

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

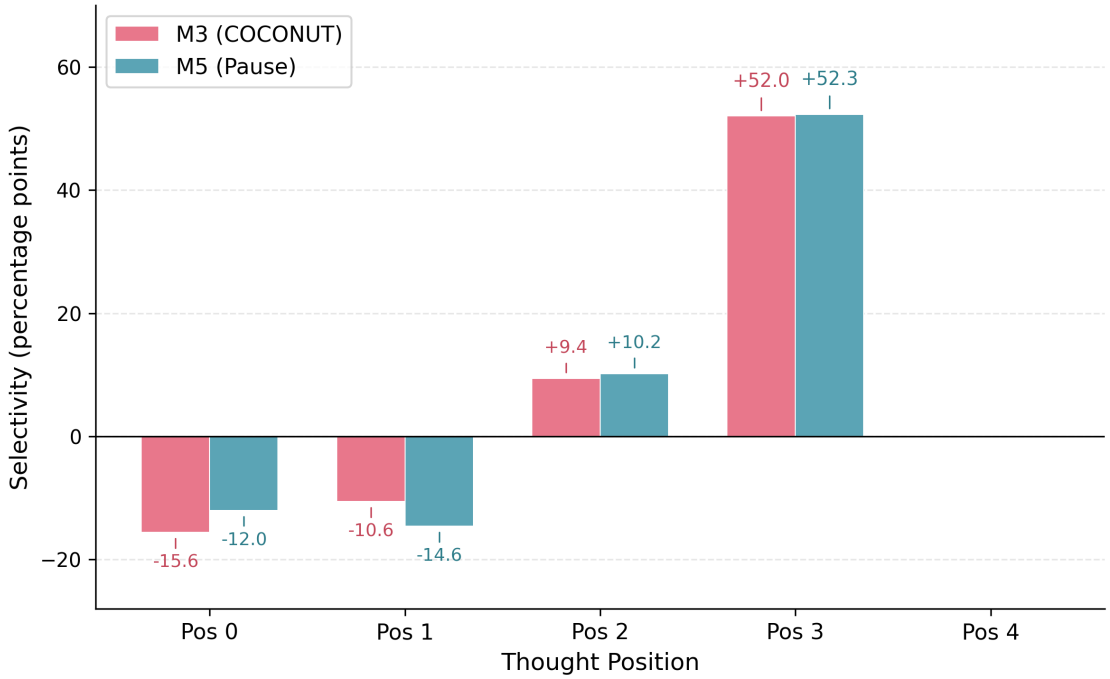

Figure 3: Figure 3: Step selectivity by thought position for M2 and M3. Both models show near-identical profiles: anti-selectivity at positions 0–1, mild selectivity at position 2, and strong selectivity at position 3 (+52.0pp and +52.3pp respectively).

The two models concentrate decodable information at different locations in the network. M2'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. M3 builds its representations through the transformer stack, with peak accuracy at layer 12 (the final layer). The diagonal peak layers for M2 are [8, 12, 12, 0] across positions 0–3; for M3 they are [8, 11, 12, 12]. These patterns reflect architectural differences in where information is injected, not differences in what information is encoded.

M2's higher thought-vs-input advantage (10.5% vs. 4.0%) and its nearly 3× greater number of significant probing cells (29/78 vs. 11/78) show that hidden-state recycling injects substantially more task-relevant information into thought positions. M2 achieves significant probe accuracy across all

layers at positions 2–3, while M3's significant cells are confined to late layers (9–12), reflecting the architectural difference in where information enters the network. However, this richer encoding does not translate to a performance advantage: M3 matches M2 on in-distribution accuracy and exceeds it on most out-of-distribution tests (Section 4.4). MLP probes with tuned hyperparameters (grid search over 72 configurations; Appendix A.7) reveal a position-dependent pattern: at position 3 (the answer hop, n = 298), linear probes outperform MLPs by ~10 percentage points due to overfitting, indicating that final-hop information is linearly separable. At position 2 (the intermediate hop, n = 500), MLPs show a +10.2pp advantage for M2 and +7.6pp for M3, revealing nonlinear structure at intermediate positions that linear probes miss. Both models show the same qualitative pattern, consistent with the shared curriculum producing similar representational strategies. The probing heatmaps for both models are shown in Figure 4.

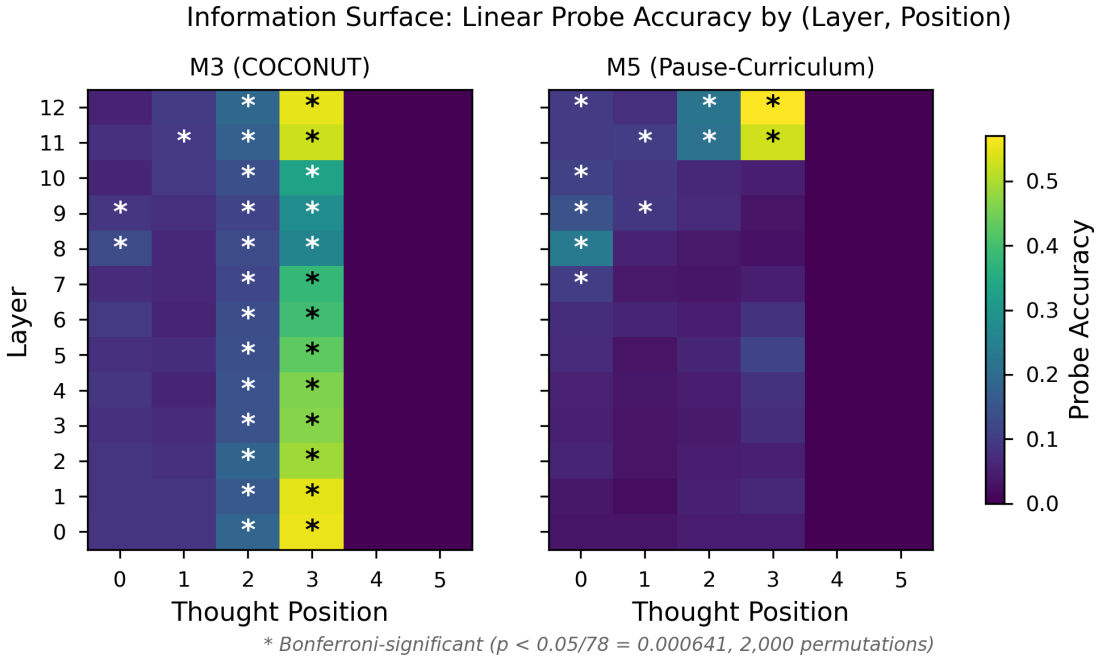

Figure 4: Figure 4: Linear probe accuracy heatmaps (layer x thought position) for M2 (COCONUT) and M3 (Pause).

4.4 Experiment 3: Out-of-Distribution Generalization

We evaluated all four 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 for all models, with pairwise comparisons between M2, M3, and M4 using exact McNemar's test.

Table 5a: Out-of-distribution accuracy for all models.

Test Set	n	M1 (CoT)	M2 (COCONUT)	M3 (Pause)	M4 (Pause-Multipass)
ProsQA (ID)	500	83.0%	97.0%	96.6%	94.8%

Test Set	n	M1 (CoT)	M2 (COCONUT)	M3 (Pause)	M4 (Pause-Multipass)
7-hop	1000	10.7%	66.0%	75.4%	76.9%
8-hop	1000	8.2%	67.5%	75.1%	75.2%
DAG	1000	28.2%	59.2%	51.9%	59.8%
Dense	1000	14.1%	61.2%	68.4%	64.8%

Table 5b: Pairwise McNemar comparisons (M3 vs. M2, retained from two-model analysis).

Test Set	M3 – M2	b	c	p (exact)	p (Bonf.)	Sig.
ProsQA (ID)	–0.4 pp	14	12	0.845	1.000	No
7-hop	+9.4 pp	120	214	< 0.001	< 0.001	Yes
8-hop	+7.6 pp	122	198	< 0.001	< 0.001	Yes
DAG	–7.3 pp	235	162	< 0.001	0.0015	Yes
Dense	+7.2 pp	139	211	< 0.001	< 0.001	Yes

Table 5c: Factorial decomposition. M4 vs. M2 isolates recycled content (same processing, different content); M4 vs. M3 isolates sequential processing (same content, different processing). McNemar comparisons recomputed using experiment-pipeline per-sample predictions for all models.

Test Set	M4 – M2	b	c	p (Bonf.)	Sig.	M4 – M3	b	c	p (Bonf.)	Sig.
ProsQA (ID)	–2.2 pp	21	10	0.354	No	–1.8 pp	19	10	0.680	No
7-hop	+10.9 pp	113	222	< 0.001	Yes	+1.5 pp	124	139	1.000	No
8-hop	+7.7 pp	111	188	< 0.001	Yes	+0.1 pp	140	141	1.000	No
DAG	+0.6 pp	176	182	1.000	No	+7.9 pp	156	235	< 0.001	Yes
Dense	+3.6 pp	150	186	0.280	No	–3.6 pp	193	157	0.306	No

The M3 vs. M2 comparisons replicate the two-model analysis: M3 outperforms M2 on 7-hop (+9.4pp), 8-hop (+7.6pp), and dense (+7.2pp), while M2 outperforms M3 on DAG (–7.3pp), all significant after Bonferroni correction. The OOD accuracy pattern is shown in Figure 5.

The factorial decomposition via M4 cleanly separates the two confounded factors. The M4 vs. M2 comparison (isolating recycled content while matching sequential processing) reveals that recycled content actively hurts chain-length extrapolation: M4 outperforms M2 by 10.9pp on 7-hop and 7.7pp on 8-hop (both $p < 0.001$ after Bonferroni correction), while showing no significant difference on DAG (+0.6pp, $p = 1.0$) or dense (+3.6pp, $p = 0.280$). The M4 vs. M3 comparison (isolating sequential processing while matching fixed embeddings) reveals that sequential processing helps topological generalization: M4 outperforms M3 by 7.9pp on DAG ($p < 0.001$), while showing no significant difference on 7-hop (+1.5pp, $p = 1.0$), 8-hop (+0.1pp, $p = 1.0$), or dense (–3.6pp, $p = 0.306$).

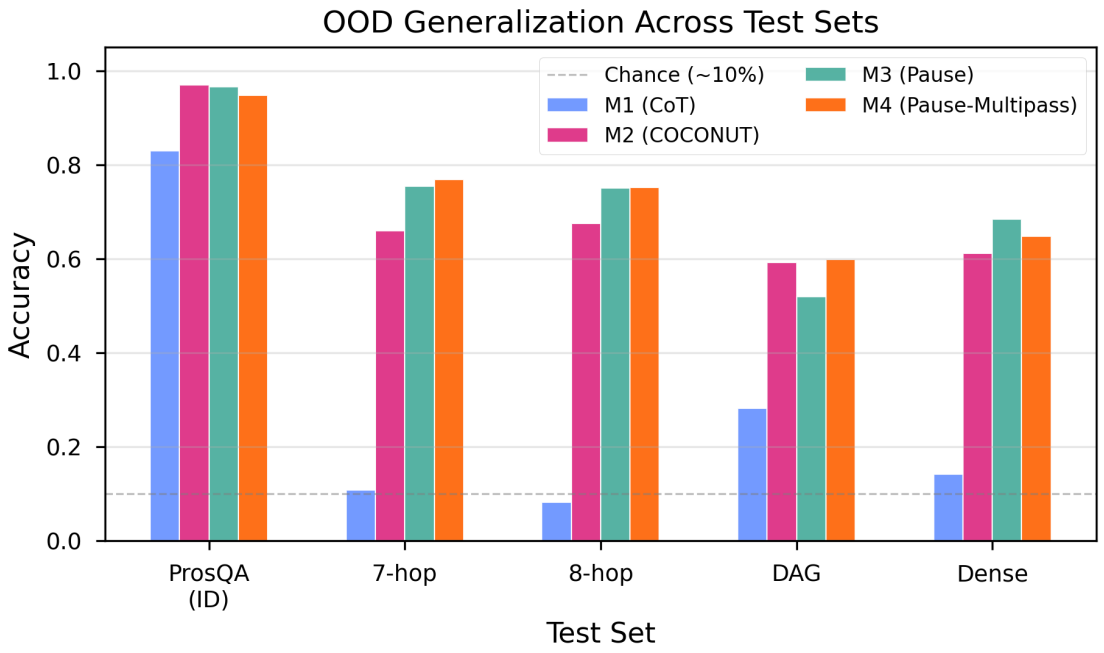

Figure 5: Figure 5: Out-of-distribution accuracy for M1, M2, M3, and M4 across four test sets.

The decomposition is approximately additive. On 7-hop, M3 outperforms M2 by 9.4pp (Table 5b). The factorial components are: recycled content costs M2 10.9pp relative to M4, and sequential processing is neutral (+1.5pp, ns). On DAG, M2 outperforms M3 by 7.3pp. The factorial components are: sequential processing gives M4 a 7.9pp advantage over M3, and recycled content is neutral (+0.6pp, ns). In both cases, the factorial effects sum to approximately the observed two-model difference, confirming that the factors combine additively rather than interacting.

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, single-pass pause tokens, or multi-pass pause tokens, provides substantial generalization benefits over explicit chain-of-thought at this model scale.

4.5 Teacher-Forced Confidence Analysis

The accuracy comparisons in Sections 4.1 and 4.4 use binary correct/incorrect judgments, discarding information about how confidently each model generates its answer. Teacher-forced log-probabilities provide a more granular measure: for each sample, we force-decode through the answer prefix ("### [Name] is a") and extract log P(species_token), where the species token is the discriminative entity requiring multi-hop graph traversal. Wilcoxon signed-rank tests on the paired log-probability differences yield per-sample confidence comparisons that reveal systematic differences invisible to binary accuracy tests.

Table 7 reports all three pairwise comparisons across five test sets. All p-values are Bonferroni-corrected within each comparison family (k = 5).

Table 7: Wilcoxon signed-rank comparisons on teacher-forced species-token log-probabilities. r = effect size (rank-biserial correlation); direction indicates which model assigns higher median

probability to the correct answer. Bonferroni-corrected (k = 5).

Comparison	Test Set	n	r	p (Bonf.)	Direction	Sig.
M2 vs. M3	ProsQA (ID)	500	0.591	< 10^{-38}	M2 > M3	Yes
	7-hop	1000	0.006	1.000	—	No
	8-hop	1000	0.006	1.000	—	No
	DAG	1000	0.003	1.000	—	No
	Dense	1000	0.113	0.002	M3 > M2	Yes
M2 vs. M4	ProsQA (ID)	500	0.678	< 10^{-50}	M2 > M4	Yes
	7-hop	1000	0.109	0.003	M2 > M4	Yes
	8-hop	1000	0.082	0.049	M2 > M4	Yes
	DAG	1000	0.073	0.106	—	No
	Dense	1000	0.118	0.001	M2 > M4	Yes
M3 vs. M4	ProsQA (ID)	500	0.286	< 10^{-9}	M3 > M4	Yes
	7-hop	1000	0.142	< 0.001	M4 > M3	Yes
	8-hop	1000	0.120	0.001	M4 > M3	Yes
	DAG	1000	0.136	< 0.001	M4 > M3	Yes
	Dense	1000	0.021	1.000	—	No

In-distribution confidence hierarchy. On the ProsQA test set, all three models achieve near-ceiling median probabilities (M2: 99.998%, M3: 99.978%, M4: 99.949%), but the paired differences are highly significant. M2 assigns systematically higher confidence than M4 (r = 0.678, a large effect), and M3 assigns higher confidence than M4 (r = 0.286, a medium effect). The M2 > M4 comparison isolates the contribution of recycled content while controlling for sequential processing: the recycled hidden states carry reasoning-relevant information that translates to measurably higher per-sample confidence, even when both models achieve comparable binary accuracy (97.0% vs. 94.8%, McNemar p_Bonf = 0.354). The absolute magnitude of this difference is small – all three models assign median probabilities above 99.9% to the correct answer on ID data – but the consistency of the paired differences produces a large rank-based effect size. This confirms the probing finding (Section 4.3) that M2's thought positions encode more task-relevant information — and demonstrates that this information is functionally accessible to the answer-generation head, not merely decodable by an external probe.

Confidence-accuracy dissociation on OOD. The most striking finding emerges on out-of-distribution chain-length tasks. On 7-hop, M2 assigns significantly higher confidence than M4 (r = 0.109, p_Bonf = 0.003), yet M2 achieves substantially lower accuracy (66.0% vs. 76.9%, McNemar +10.9pp, p_Bonf < 0.001). The same pattern appears on 8-hop: M2 is more confident (r = 0.082, p_Bonf = 0.049) but less accurate (67.5% vs. 75.2%, +7.7pp). This confidence-accuracy dissociation indicates that recycled hidden states make COCONUT overconfident on out-of-range problems: the distribution-specific information encoded in the recycled content produces high-confidence predictions that are systematically wrong. In contrast, M2 and M3 show no confidence difference on 7-hop or 8-hop (r < 0.01, p = 1.0), consistent with their indistinguishable performance on these test sets when no recycled content is present to create miscalibration.

Sequential processing and OOD confidence. The M3 vs. M4 comparison reveals that sequential processing (controlling for content) increases OOD confidence: M4 assigns higher median probabilities than M3 on 7-hop (r = 0.142), 8-hop (r = 0.120), and DAG (r = 0.136), all significant. On DAG, this higher confidence aligns with M4's higher accuracy (+7.9pp). On 7-hop and

8-hop, M4's slightly higher accuracy (+1.5pp and +0.1pp, both ns by McNemar) is accompanied by significantly higher confidence, suggesting that sequential processing produces better-calibrated representations on OOD tasks — a benefit that is modest in accuracy terms but substantial in confidence terms.

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 sequential reasoning hypothesis and the curriculum-driven computation hypothesis make divergent predictions, the data favor curriculum-driven computation. Table 6 summarizes the alignment. We note that the evidence does not support pure "buffering" in the sense of unstructured generic computation: the strong selectivity at position 3 (+52pp) and anti-selectivity at positions 0–1 reveal structured, position-specific encoding in both models. However, this structure arises from the shared curriculum rather than from the recycling mechanism, and it does not constitute the sequential step-by-step reasoning chain that the COCONUT architecture was designed to enable.

Table 6: Summary of convergent evidence across experimental paradigms.

Evidence	Sequential reasoning prediction	Curriculum-driven prediction	Our result
Permutation sensitivity	Order matters	Order irrelevant (for output)	0% flip rate for both M2 and M3
Cross-transplant	Problem-specific states	Generic / curriculum-shaped	Both tolerate foreign thoughts (M2: 97.0%, M3: 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	Sequential reasoning prediction	Curriculum-driven prediction	Our result
Probing selectivity	M2-specific step encoding	Shared curriculum-driven pattern	Both models show identical selectivity profiles: strong step-specificity at position 3 (+52pp), anti-selectivity at positions 0–1, arising from shared curriculum
Probing significance	M2 broadly significant	Equal significance	M2: 29/78 significant cells; M3: 11/78. COCONUT encodes more broadly but without behavioral gain
Thought-vs-input advantage	Only COCONUT benefits	Equal benefit	M2 higher (10.5% vs. 4.0%), but does not produce different strategy

Evidence	Sequential reasoning prediction	Curriculum-driven prediction	Our result
OOD generalization	COCONUT advantages	Equal or M3 advantages	M3 wins 3/4; M2 wins DAG (all significant). Factorial decomposition: recycled content hurts chain-length extrapolation; sequential processing drives DAG advantage (Section 5.3)
Teacher-forced confidence	M2 uniformly more confident	Equal confidence	M2 more confident on ID ($r = 0.678$) but overconfident on OOD: higher confidence with lower accuracy on 7-hop and 8-hop (Section 4.5)

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, seven independent diagnostics consistently fail to find evidence that COCONUT's recycled hidden states carry reasoning content that differs functionally from M3'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: M2 and M3 show a task-dependent generalization tradeoff, which M4's factorial decomposition fully resolves — attributing M2's chain-length disadvantage to recycled content and M2's DAG advantage to sequential processing (Section 5.3). The teacher-forced confidence analysis (Section 4.5) adds a further dimension: M2's recycled content produces significantly

higher per-sample confidence on in-distribution data, confirming that the recycled hidden states carry reasoning-relevant information. However, this higher confidence becomes miscalibrated on OOD chain-length tasks, where M2 is simultaneously more confident and less accurate than M4 — the recycled content does not merely fail to help on longer chains, it actively misleads.

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 M2 and M3, 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.

M2's thought-token positions encode 10.5% more decodable information than its input positions, compared with 4.0% for M3. Corrected permutation tests (2,000 permutations, Bonferroni-corrected) confirm this quantitative asymmetry: M2 achieves 29/78 significant probing cells versus 11/78 for M3. All 13 layers are significant at M2 positions 2 and 3, while M3's significant cells are confined to late layers (9–12). The recycling mechanism creates broadly distributed, robustly decodable intermediate representations, consistent with its architectural design. But this richer encoding does not produce a different selectivity pattern or an accuracy advantage: M3 matches M2 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.

M3's selectivity pattern at position 3 is likely mediated by GPT-2's learned positional encodings, which provide the only position-distinguishing signal in M3's otherwise identical thought-token sequence. The curriculum trains both models to route final-hop information to position 3 – in M3'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 – M2'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. M3 builds its (smaller) probing signal through the standard transformer computation, peaking at layer 12, yet reaches comparable or superior accuracy.

5.3 Factorial Decomposition of OOD Performance

The two-model OOD results (M2 vs. M3) revealed a task-dependent generalization tradeoff: M3 outperforms M2 on 7-hop (+9.4pp), 8-hop (+7.6pp), and dense graphs (+7.2pp), while M2 outperforms M3 on DAGs (–7.3pp). This pattern was ambiguous because M2 and M3 differ in two confounded ways: (1) the content of thought-token embeddings and (2) the sequential processing structure. M4 resolves this ambiguity by matching M2's sequential processing while using M3's fixed embeddings.

The factorial decomposition reveals a striking task-dependent dissociation between the two confounded factors.

Chain-length extrapolation (7-hop, 8-hop). M4 matches M3, not M2. On 7-hop, M4 achieves 76.9% — nearly identical to M3's 75.4% (McNemar p_Bonf = 1.0) and significantly higher than M2's 66.0% (+10.9pp, p_Bonf < 0.001). The same pattern holds on 8-hop (M4: 75.2%, M3: 75.1%, M2: 67.5%). Since M4 and M2 share the same sequential processing structure but differ in content, the 10.9pp M4 advantage on 7-hop is entirely attributable to the absence of recycled content. Since M4 and M3 share fixed embeddings but differ in processing structure, the 1.5pp M4 advantage (ns) confirms that sequential processing does not affect chain-length extrapolation. The recycled hidden states actively impede generalization to longer chains, possibly because they carry distribution-specific information calibrated to the training range of 3–6 hops that becomes misleading at longer path lengths.

Topological generalization (DAG). M4 matches M2, not M3. On DAG, M4 achieves 59.8% — nearly identical to M2's 59.2% (McNemar p_Bonf = 1.0) and significantly higher than M3's 51.9% (+7.9pp, p_Bonf < 0.001). The 7.9pp advantage is entirely attributable to sequential processing: both M4 and M2 process thought tokens across 6 sequential passes via KV-cache incremental decoding, while M3 processes all tokens in a single pass. The sequential structure provides an inductive bias that helps with novel graph topologies — the forced step-by-step accumulation of information across passes may implicitly encourage a search strategy better suited to DAG structures than M3's parallel processing. Critically, this advantage comes from the processing structure, not from the recycled content: M4 achieves it with fixed pause embeddings.

Dense graphs. Neither factor alone reaches significance. M4 (64.8%) falls between M2 (61.2%) and M3 (68.4%), with M4–M2 = +3.6pp (p_Bonf = 0.280) and M4–M3 = –3.6pp (p_Bonf = 0.306). The opposing directions suggest that both factors may contribute — recycled content slightly hurts, sequential processing slightly hurts — but neither effect is individually resolvable at this sample size. Dense graphs may require a different set of computational primitives than either chain extension or topological navigation.

Summary. The two-model OOD comparison (M3 vs. M2) revealed a generalization tradeoff that was attributionally ambiguous. The factorial decomposition resolves this completely: M2's chain-length disadvantage arises from recycled content (not processing structure), and M2's DAG advantage arises from sequential processing (not recycled content). The recycled hidden states are not merely inert — on chain-length extrapolation, they are actively harmful.

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 M2 and M3 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 M3 baseline confirms and extends this finding: curriculum-trained pause tokens match COCONUT on in-distribution ProsQA (96.6% vs 97.0%, $p = 0.845$) and outperform it 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 M2 and M3 share, and it is the component that separates both models from the M1 chain-of-thought baseline by 13.6-14.0 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 0.4-percentage-point test-set gap between M2 (97.0%) and M3 (96.6%) is not statistically significant (McNemar $p = 0.845$), but could widen or reverse under different random initializations. The out-of-distribution advantages we report for M3 – 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. Training-time evaluation at best epoch yielded a larger apparent gap (M2 = 98.0%, M3 = 95.6%), differing from the experiment-pipeline numbers by 5 samples per model; this sensitivity to inference implementation underscores the need for multi-seed replication.

Forward-pass asymmetry (addressed by M4). M2 processes thought tokens sequentially via KV-cache incremental decoding, while M3 processes all thought tokens in a single forward pass. In the two-model comparison, this meant M2 and M3 differed in two confounded ways: (1) the *content* of thought-token embeddings and (2) the *sequential processing structure*. M4 resolves this confound by matching M2's sequential processing while using M3's fixed embeddings. The factorial decomposition (Section 5.3) shows that the two factors contribute independently and to different OOD tasks: recycled content hurts chain-length extrapolation (M4 outperforms M2 on 7-hop and 8-hop), while sequential processing helps topological generalization (M4 outperforms M3 on DAG). While M4 resolves the content-vs-processing confound, a residual asymmetry remains: M4 and M2 process the same number of sequential steps, but the information available at each step differs qualitatively (fixed embedding vs. a representation that reflects accumulated state). This is an inherent property of the recycling mechanism and cannot be further decomposed without more invasive interventions.

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. M2's 10.5% mean thought-position advantage over input positions, and its nearly 3× greater number of significant probing cells (29/78 vs. 11/78), demonstrate that the recycling mechanism has a measurable and substantial effect on the model's internal representations. The mechanism is not "doing nothing" – it injects broadly distributed, decodable information that the pause baseline does not contain. Our claim is narrower: this richer encoding 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 M2 vs. 4.09 for M3), reflecting the 50-fold variance difference between recycled hidden states and near-constant pause embeddings. Our cross-scale analysis (applying M2-magnitude noise to M3) 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.

M4 experimental coverage. The corruption analysis and representational probing experiments could not be extended to M4 due to a methodological incompatibility with multi-pass KV-cache architectures. Both experiments extract hidden states by running a fresh forward pass on the model's input embeddings, which discards the accumulated KV-cache state from M4's 6-pass sequential processing. For M4, the KV-cache accumulated across passes IS the model's computation — without it, the extracted representations do not reflect M4's inference-time behavior. Corruption injection and probing on these artifacts would measure properties of the extraction pipeline, not of the model. Extending these experiments to M4 would require per-pass hidden state collection from within the KV-cache loop, which we leave for future work. The factorial decomposition of OOD performance and the Wilcoxon confidence analysis (Sections 4.4 and 4.5) provide the primary evidence for M4's behavior.

7 Conclusion

We asked whether COCONUT's continuous thought tokens perform sequential latent reasoning or serve primarily as curriculum-shaped computational scaffolding. A curriculum-matched pause-token baseline (M3), trained under COCONUT's own 7-stage curriculum, matches COCONUT on in-distribution ProsQA (96.6% vs 97.0%, McNemar $p = 0.845$) without recycling any hidden states. A second control (M4) matches COCONUT's sequential multi-pass processing structure while using fixed pause embeddings, enabling factorial decomposition of the recycled-content and sequential-processing factors. Three converging experiments – corruption analysis, representational probing, and cross-model transplantation – fail to distinguish M2 and M3 on any diagnostic where sequential reasoning and curriculum-driven computation make divergent predictions. Both models exhibit structured, position-specific representations – including strong step-selectivity at position 3 and a broadcast pattern at early positions – but these patterns arise from the shared curriculum rather than from the recycling mechanism.

On out-of-distribution generalization, the two-model comparison (M2 vs. M3) reveals a task-dependent tradeoff: M3 outperforms COCONUT on 3 of 4 test sets, while COCONUT holds a significant advantage on DAG structures. The factorial decomposition via M4 fully resolves this ambiguity: M2's chain-length disadvantage is caused by the recycled content (M4 outperforms M2 by 10.9pp on 7-hop, $p < 0.001$), not the processing structure (M4 and M3 do not differ). M2's DAG advantage is caused by the sequential processing structure (M4 outperforms M3 by 7.9pp on DAG, $p < 0.001$), not the recycled content (M4 and M2 do not differ). The recycled hidden states are not merely inert — on chain-length extrapolation, they are actively harmful, potentially because they carry distribution-specific information that misleads the model on out-of-range inputs.

These results indicate that COCONUT's in-distribution accuracy on ProsQA is primarily attributable to its training curriculum, not to the content of the recycled hidden states. The re-

cycling mechanism does have a measurable effect — it produces significantly higher per-sample confidence (Wilcoxon r = 0.678 on in-distribution data) and more broadly distributed probing signal (29/78 vs. 11/78 significant cells) — but this richer encoding does not translate to higher accuracy. On OOD chain-length tasks, the higher confidence becomes actively miscalibrated, producing overconfident wrong answers (Section 4.5). The curriculum – which progressively removes explicit chain-of-thought tokens and forces the model to internalize multi-step computation – is the shared factor among M2, M3, and M4, and it is the factor that separates all three from the chain-of-thought baseline. For researchers developing latent reasoning architectures at this scale, this work suggests that curriculum design warrants at least as much attention as the choice of thought-token mechanism. The simpler pause mechanism achieves comparable in-distribution accuracy at lower computational cost. Code, configurations, and experiment scripts are available at https://github.com/bmarti44/research-pipeline.

References

Deng, Y., Yu, Y., Saha, S., Lu, J., & Hajishirzi, H. (2024). From explicit CoT to implicit CoT: Learning to internalize CoT step by step. *arXiv preprint arXiv:2405.14838*.

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.

Saparov, A., & He, H. (2022). Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR 2023)*.

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)*.

Zelikman, E., Harik, G., Shao, Y., Jayasiri, V., Haber, N., & Goodman, N. D. (2024). Quiet-STaR: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*.

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)*.

Appendix

A.1 Selectivity Computation Correction and Corrected Permutation Tests

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.

The same truncation invalidated the permutation-based significance tests computed during the original probing run: with only 12 samples across 38+ classes, probes returned near-zero accuracy regardless of label permutation, yielding uniformly non-significant p-values (p = 1.0 for all 78 cells).

We reran permutation tests with corrected sample sizes using 2,000 permutations per cell (minimum achievable p = 1/2001 = 0.0005, below the Bonferroni threshold of 0.05/78 = 0.000641). The permutation test uses an optimized ridge classifier with precomputed Cholesky decomposition on a single 80/20 stratified split, counting exceedances with the conservative estimator p = (count + 1) / (n_perms + 1).

M2 (COCONUT): 29/78 significant cells. All 13 layers are significant at positions 2 and 3 (26 cells), plus layer 8 and 9 at position 0, and layer 11 at position 1. The peak accuracy of 55.4% at (layer 0, position 3) achieves p = 0.0005. Every layer at position 3 exceeds 25% accuracy (38-class chance = 2.6%), and every layer at position 2 exceeds 11.8%.

M3 (Pause): 11/78 significant cells. Significant cells are concentrated in late layers: layers 7–10 and 12 at position 0, layers 9 and 11 at position 1, layers 11–12 at position 2, and layers 11–12 at position 3. The peak accuracy of 57.0% at (layer 12, position 3) achieves p = 0.0005.

The key difference is in the distribution of significant cells. M2 shows significant probing accuracy across all layers at positions 2–3, consistent with the recycling mechanism injecting decodable information from the earliest layer. M3 shows significance only in late layers (primarily 9–12), consistent with representations being built through the transformer stack. This architectural difference in where information is available does not produce a behavioral difference: both models achieve comparable task accuracy and selectivity profiles.

Positions 4 and 5 return 0.0% accuracy for both models (n = 81 with 32 classes and n = 12 with 12 classes, respectively); with more classes than the minimum fold size, stratified cross-validation cannot be computed, and these cells are excluded from significance testing.

A.2 Cross-Corruption Results

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

Positions Corrupted	M2 + M2-noise (L2~203)	M3 + M3-noise (L2~4)	M3 + M2-noise (L2~203)
0 (clean)	97.0%	96.6%	96.6%
1	96.8%	96.4%	96.6%

Positions Corrupted	M2 + M2-noise (L2~203)	M3 + M3-noise (L2~4)	M3 + M2-noise (L2~203)
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	M2	M3
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	M2	M3
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	M2	M3
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%

Position Corrupted	M2	M3
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 carry mutually redundant copies of answer-relevant content.

A.6 Full Linear Probe Accuracy Grids

Table A5: M2 (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: M3 (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

Layer	Pos 0	Pos 1	Pos 2	Pos 3	Pos 4	Pos 5
12	10.2	7.8	22.0	57.0	0.0	0.0

A.7 Nonlinear Probe Results

Our initial MLP probes (2-layer, 256 hidden units, scikit-learn MLPClassifier with default hyperparameters) produced a uniform 0/78 null result. As anticipated in our original caveat, this reflected convergence failure rather than a genuine absence of nonlinear encoding.

Grid search methodology. We conducted a systematic hyperparameter search over the five cells with highest linear probe accuracy: M2 (layer 0, position 3), M2 (layer 12, position 2), M3 (layer 12, position 3), M3 (layer 8, position 0), and M3 (layer 12, position 2). For each cell, we evaluated 72 configurations: 6 hidden layer sizes (64, 96, 128, 192, 256, 512) $\times$ 3 learning rates (0.0001, 0.001, 0.01) $\times$ 4 L2 regularization strengths (0.0001, 0.001, 0.01, 0.1), using 5-fold cross-validation and max_iter=2000. All probes used the cached hidden states from the corrected probing analysis.

MLP probe grid search results at five target cells (advantage = best MLP accuracy − linear probe accuracy):

Model	Layer	Pos	N	Linear	Best MLP	Advantage
M2	0	3	298	55.4%	46.0%	–9.4pp
M2	12	2	500	19.0%	29.2%	**+10.2pp**
M3	12	3	298	57.0%	45.6%	–11.4pp
M3	8	0	500	23.6%	14.6%	–9.0pp
M3	12	2	500	22.0%	29.6%	**+7.6pp**

The results reveal a position-dependent pattern. At **position 3** (the answer hop), linear probes outperform MLPs by approximately 10 percentage points. With n = 298 samples and 38 target classes (~7.8 samples per class), the MLP's larger parameter count overfits despite regularization. The information encoded at position 3 is linearly separable – consistent with a broadcast representation where the final-hop entity is placed in a linearly accessible format for the answer-generation head.

At **position 2** (the intermediate hop), MLPs show a substantial advantage: +10.2 percentage points for M2 and +7.6 for M3. With n = 500 samples, overfitting is less severe, and the MLP captures nonlinear structure that linear probes miss. This suggests that intermediate reasoning steps are encoded in a more complex, nonlinearly distributed format, while the final answer is projected into a linearly decodable subspace.

Both models show the same qualitative pattern (linear-sufficient at position 3, nonlinear advantage at position 2), consistent with the shared curriculum producing similar representational strategies. The MLP advantage at position 2 does not alter the main finding that both architectures achieve near-identical step selectivity at position 3 (Section 4.3).

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

A.9 Datasets

Examples We provide an example of the questions and chain-of-thought solutions for ProsQA, the dataset used in our experiments.

Question: "Every shumpus is a rempus. Every shumpus is a yimpus. Every terpus is a fompus. Every terpus is a gerpus. Every gerpus is a brimpus. Alex is a rempus. Every rorpus is a scrompus. Every rorpus is a yimpus. Every terpus is a brimpus. Every brimpus is a lempus. Tom is a terpus. Every shumpus is a timpus. Every yimpus is a boompus. Davis is a shumpus. Every gerpus is a lorpus. Davis is a fompus. Every shumpus is a boompus. Every shumpus is a rorpus. Every terpus is a lorpus. Every boompus is a timpus. Every fompus is a yerpus. Tom is a dumpus. Every rempus is a rorpus. Is Tom a lempus or scrompus?"

Reasoning steps: ["Tom is a terpus.", "Every terpus is a brimpus.", "Every brimpus is a lempus."]

Answer: "Tom is a lempus."

The input contains 23 statements, of which only 3 are relevant to the answer. The remaining 20 are valid inheritance rules involving other entities (Alex, Davis) and concept chains that serve as distractors.

Construction of ProsQA ProsQA is constructed following the methodology of Hao et al. (2024), which draws on the entity-concept naming conventions of ProntoQA (Saparov and He, 2022). Each problem is structured as a binary question: "Is [Entity] a [Concept A] or [Concept B]?" where [Concept A] is the correct answer.

The underlying graph is a directed acyclic graph (DAG) where each node represents an entity or a concept. The graph is constructed such that a path exists from [Entity] to [Concept A] but not to [Concept B]. New nodes are incrementally added and randomly connected with edges. With probability 0.35, the new node is constrained to not be a descendant of node 1 (the correct-answer branch); with probability 0.35, it is constrained to not be a descendant of node 0 (the entity); otherwise, it may connect to any existing node. This separation maintains distinct families of nodes and balances their sizes to prevent model shortcuts. Sampling weights prioritize deeper nodes (weight = depth $\times$ 1.5 + 1) to encourage longer reasoning chains. The number of incoming edges per new node is drawn from Poisson(1.5).

After graph construction, root nodes (those without parents) are assigned entity names (e.g., "Tom," "Alex"), while other nodes receive fictional concept names (e.g., "brimpus," "lorpus"). Node 0 becomes the queried [Entity], a leaf labeled 1 becomes [Concept A], and a leaf labeled 2 becomes [Concept B]. The two answer options are randomly permuted to avoid positional biases.

Table A8: Statistics of the ProsQA graph structure (from Hao et al., 2024).

# Nodes	# Edges	Len. of Shortest Path	# Shortest Paths
23.0	36.0	3.8	1.6

Table A9: Dataset split sizes. We use ProsQA exclusively in this work.

Dataset	Training	Validation	Test
ProsQA	17,886	300	500