Towards Inference-time Scaling for Continuous Space Reasoning

Minghan Wang[1], Thuy-Trang Vu[1], Ehsan Shareghi[2], Gholamreza Haffari[1]
[1]Department of Data Science & AI, Monash University
[2]Department of Computer Science, University College London
{minghan.wang,trang.vu1,gholamreza.haffari}@monash.edu
ehsan.shareghi@ucl.ac.uk

Abstract

Inference-time scaling through multiple sample generation in combination with Process- or Outcome-Reward Model (PRM or ORM) reranking has proven effective for text-based reasoning in large language models. This paper investigates whether such established techniques can be successfully adapted to reasoning in the continuous space, using COCONUT (Hao et al., 2024) continuous space reasoning LM as the backbone. We demonstrate the feasibility of generating diverse reasoning paths through dropout-based sampling. Our Pass@N analysis on the generated samples reveals the potential that could enable a significant gain in performance akin to gains observed in the discrete space. However, we highlight unique challenges faced for materializing this gain in the continuous thought space. In particular, working recipes for data generation and training PRM and ORM models in the discrete space unlocks only marginal improvements in the continuous space. Through probing various aspects including geometric properties and trajectory dynamics, we identify the underlying reasons that prevent effective discrimination between correct and incorrect reasoning (essential for the functioning of PRM and ORM). Our findings reveal that current limitations stem from the absence of key inductive biases in continuous thought representations. [1]

1 Introduction

Recent advances in reasoning large language models have expanded along training and inference-time dimensions. While training paradigms continue evolving (OpenAI et al., 2024; DeepSeek-AI et al., 2025), inference-time scaling has stabilized around generating multiple samples per input and employing Process/Outcome Reward Models (PRM/ORM) for re-ranking candidates (Uesato et al., 2022; Lightman et al., 2023; Wang et al., 2024, 2023) and delivered substantial accuracy improvements across reasoning tasks.

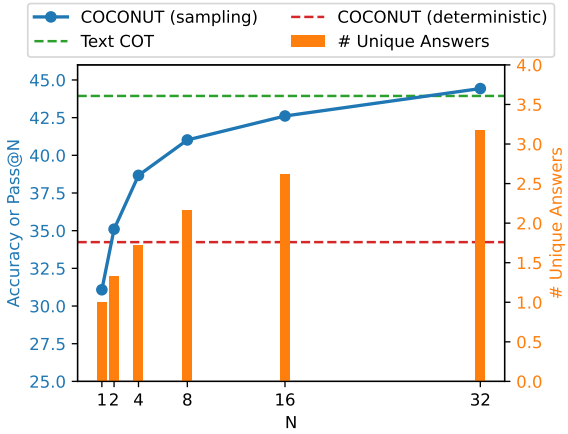

Figure 1: Performance comparison of COCONUT sampling vs. baselines on GSM8k. The blue line shows Pass@N performance with dropout-based sampling, demonstrating substantial potential for inference-time scaling compared to deterministic COCONUT (red dashed line) or text CoT (green dashed line). Orange bars indicate the average number of unique answers generated at each sample size, showing logarithmic growth that suggests efficient reranking is feasible.

Continuous reasoning (Hao et al., 2024; Dehghani et al., 2019; Mohtashami et al., 2024; Cheng and Durme, 2024) has emerged as an alternative paradigm that trades interpretability for computational efficiency. Unlike Chain-of-Thought (CoT) methods generating explicit textual steps (Wei et al., 2023), continuous reasoning models like COCONUT (Hao et al., 2024) operate directly in latent space, enabling faster inference while maintaining multi-step reasoning capabilities at the cost of transparency. This work investigates whether established inference-time scaling techniques from discrete reasoning can be effectively transferred to continuous reasoning models. Specifically, we explore enabling inference-time scaling through multiple sample generation in continuous space and de-

[1]Our code is available at: https://github.com/yuriak/LatentITS

Findings of the Association for Computational Linguistics: ACL 2026, pages 26842–26856

veloping reliable reward models to rank continuous thoughts. Using COCONUT as our experimental backbone, we demonstrate that dropout-based sampling can generate diverse reasoning paths, with Pass@N analysis revealing substantial potential to improve overall reasoning accuracy (as shown in Figure 1) in mathematical reasoning. This indicates a significant room for improvement through effective ranking mechanisms.

Our attempts to train PRM/ORM models by adapting established automatic annotation protocols from discrete reasoning, such as MATH-Shepherd (Wang et al., 2024), yield limited gains and fall short of the improvements observed in discrete settings. Through a detailed analysis of continuous thoughts, we find that this limitation arises from geometric homogeneity in the latent space, which hinders reliable discrimination between *correct* and *incorrect* reasoning trajectories. To validate this hypothesis, we further introduce a lightweight post-training intervention that injects explicit geometric inductive bias, showing that even a minimal adjustment can partially restore separability and improve inference-time scaling, while indicating the need for more principled training objectives for continuous reasoning models.

2 Related Work

2.1 Inference-Time Scaling in Discrete Space

Inference-time scaling aims to enhance the reasoning capability of LLMs at test time by generating and selecting among multiple reasoning paths, without modifying model parameters. Recent works (Lightman et al., 2023; Li et al., 2025) have outlined this as a key direction for enabling more reliable and deliberative reasoning in LLMs.

One foundational strategy is Majority voting (a.k.a. self-consistency) (Wang et al., 2023), which improves reasoning by sampling multiple Chain-of-Thoughts and selecting the most consistent answer by picking the most frequent answer. A natural extension of self-consistency is best-of-N sampling with reranking, where multiple reasoning paths are sampled at test time and the final answer is chosen based on scoring criteria such as likelihood, confidence, or verifier output. These methods include structured prompting approaches such as Least-to-Most (Zhou et al., 2023), and iterative refinement or search-based techniques like Self-Refine (Madaan et al., 2023) and Tree-of-Thoughts (Yao et al., 2023), illustrating the range of reranking mechanisms across different granularity levels.

Another line of inference-time methods employs verifier models to evaluate and rank sampled reasoning paths based on their correctness or plausibility, often at the step level rather than just final answers. To assess the correctness of these intermediate steps, PRMs have been introduced to assign a score to reasoning steps, reflecting their correctness likelihood. One of the key challenges in training such models is the lack of step-level human annotations (with the exception of PRM800k dataset (Lightman et al., 2023)), which is often circumvented via automatic annotation methods. Among existing methods, Math-Shepherd (Wang et al., 2024) is commonly used as a de-facto approach for automatically generating step-level supervision from existing CoT outputs. There are other approaches that propose alternative means for more reliable supervision (Zhang et al., 2025).

2.2 Continuous Space Reasoning

Continuous CoT performs multi-step inference directly in a model's latent space, bypassing explicit textual steps (Sui et al., 2025). Instead of generating intermediate tokens, the model refines hidden representations across steps, aiming for more efficient and expressive reasoning (Zhu et al., 2025). Several methods implement this idea. CODI (Shen et al., 2025) improves latent stability by aligning hidden states between student and teacher models using self-distillation. CCOT (Cheng and Durme, 2024) compresses multi-step reasoning into a variable-length sequence of latent embeddings, optionally decodable for interpretability. Token Assorted (Su et al., 2025) introduces a hybrid approach that combines discrete latent tokens generated by VQ-VAE with text tokens. CoT2 (Gozeten et al., 2025) demonstrates that models are able to track multiple traces in parallel when reasoning in continuous space and proposes strategies to explicitly encourage such parallel reasoning.

Among these methods, COCONUT (Hao et al., 2024) represents a prominent work in continuous reasoning that enables models to perform reasoning directly in hidden state space rather than through explicit textual steps. During training, the model employs multi-stage curriculum learning where, at training stage k, the first k reasoning steps are replaced with $k \times c$ continuous thought vectors (where c controls the number of latent thoughts per reasoning step), progressively transitioning from language-based to latent-based reasoning. During

inference, the reasoning process can be formalized as follows: given a problem prompt X, a special <bot> token initiates continuous reasoning mode. The model generates a sequence of continuous thought vectors $\{\mathbf{s}_1, ..., \mathbf{s}_T\}$ through autoregressive computation:

$$\mathbf{s}_i = f_\theta(X, \mathbf{s}_{<i}) \quad (1)$$

where f_θ represents the model's forward pass. Crucially, these hidden states are directly fed as input for subsequent reasoning steps without decoding to text space. After T predetermined reasoning steps, an <eot> token terminates the continuous phase, and the model returns to standard text generation to produce the final answer. In this work, we adopt COCONUT as our backbone to systematically investigate whether established inference-time scaling techniques in the text space can be effectively transferred to continuous reasoning models.

3 Generating Continuous Thought Samples

Inference-time scaling in text-based LLMs relies on sampling multiple reasoning trajectories from token probability distributions at each generation step, ensuring trajectory diversity while preserving reasoning coherence for techniques like self-consistency voting.

COCONUT presents a fundamental challenge: its reasoning process operates deterministically within continuous space. While sampling can be applied during final answer generation, all samples originate from identical reasoning trajectories, failing to achieve the path diversity necessary for effective inference-time scaling.

To address this limitation, we introduce a simple yet effective approach that injects controlled stochasticity into the continuous reasoning process. We selectively enable dropout during the iterative hidden state generation phase while disabling it during the text generation phase for answer production. This design confines randomness to the reasoning process without compromising final answer generation integrity.

3.1 Preliminary Experiments

Experimental Setup We strictly follow the original paper (Hao et al., 2024) to reproduce experiments on GSM8k-aug (Cobbe et al., 2021; Deng et al., 2024) (385k for training, 1319 for testing). Specifically, we use GPT-2 (Radford et al., 2019) as the base model and train for 6 epochs in the initial stage and 3 epochs in each of the remaining stages (3 stages total). During inference, we employ $T = 3 \times c$ where $(c = 2)$ continuous thought steps, identical to the configuration in the original paper.

For dropout-based sampling, we use the same dropout rate (0.1) as during training and enable dropout only during continuous reasoning. We evaluate with sample sizes of $\{1, 2, 4, 8, 16, 32\}$. We establish two baselines: text CoT and the standard deterministic COCONUT reasoning, reporting their accuracy scores. For the sampling evaluation, we report Pass@N metrics where N equals the sample size, i.e. a problem is considered correct if any of the N sampled answers is correct, representing the upper-bound achievable by the model. We also report the number of unique answers after deduplicating the N candidates.

Experimental Results Figure 1 reveals that deterministic COCONUT reasoning exhibits nearly a 10-point accuracy gap compared to text CoT, consistent with the original COCONUT paper. In our dropout-based sampling evaluation, Pass@1 is the single-sample accuracy and is slightly lower than deterministic COCONUT reasoning, indicating that enabling dropout during inference introduces a modest degradation in reasoning quality. Despite the performance impact at $N = 1$, Pass@N rapidly surpasses the deterministic COCONUT baseline as sample size increases, ultimately exceeding the text CoT baseline at $N = 32$. Critically, the number of unique answers exhibits logarithmic rather than linear growth with increasing N. This observation suggests that effective re-ranking methods could achieve substantial accuracy gains by identifying correct answers from the generated candidate set, while maintaining computational efficiency since re-ranking costs do not scale linearly with sample size if applying additional deduplication.

These preliminary findings motivated us to further explore the possibility of training the reward models to fully exploit the sampled solutions via re-ranking.

4 Reward Modeling for Continuous Thought

Building on our preliminary findings that demonstrate significant potential for inference-time scaling in continuous reasoning, we address the critical challenge of effectively ranking model-generated candidates. We adopt the well-established *dis-*

Algorithm 1 Thoughts Annotation

Require: Problem prompt X, ground truth answer a^*, number of trajectories M, number of MC completions N, reasoning steps T

Ensure: Step-wise rewards $\{y_{s_i}^{SE}, y_{s_i}^{HE}\}_{i=1}^{|\mathcal{T}_{unique}|}$ for all reasoning trajectories

1: **Generate reasoning trajectories:**
2: $\mathcal{T} \leftarrow \{\tau_1, \ldots, \tau_M\}$ where $\tau_m = \{\mathbf{s}_1^{(m)}, \ldots, \mathbf{s}_T^{(m)}\}$
3: Each $\mathbf{s}_i^{(m)} \in \mathbb{R}^D$ is a continuous thought vector from COCONUT
4: **Deduplicate trajectories:**
5: $\mathcal{T}_{unique} \leftarrow$ Deduplicate($\mathcal{T}$) based on final answers
6: **Annotate each reasoning step:**
7: **for** each trajectory $\tau \in \mathcal{T}_{unique}$ **do**
8: **for** $i = 1$ to T **do**
9: $\mathbf{s}_i \leftarrow \tau[i]$ {Current reasoning step}
10: $\tau_{1:i} \leftarrow \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_i\}$ {Partial trajectory}
11: **Monte Carlo estimation:** {Generate N completions from $\tau_{1:i}$}
12: $\{a_j\}_{j=1}^N \leftarrow$ Complete($\tau_{1:i}, N$)
13: $y_{s_i}^{HE} \leftarrow \mathbf{1}[\exists j : a_j = a^*]$ {Hard estimation}
14: $y_{s_i}^{SE} \leftarrow \frac{\sum_{j=1}^N \mathbf{1}[a_j = a^*]}{N}$ {Soft estimation}
15: **end for**
16: **end for**
17: **return** Step-wise reward labels for PRM training

crete data annotation framework from MATH-Shepherd (Wang et al., 2024) to construct training data for continuous thought process supervision and develop both process reward models (PRM) and outcome reward models (ORM) specifically designed for COCONUT. We examine their effectiveness in identifying and ranking correct reasoning trajectories.

4.1 Data Curation

MATH-Shepherd introduces an automated data annotation methodology using Monte Carlo (MC) estimation that circumvents expensive human annotation by estimating reasoning step success probabilities through multiple sampling. We adapt this framework to annotate continuous thought vectors for reward model (RM) training.

The annotation process can be found in Algo 1. Given a problem prompt X, we employ the trained COCONUT model to generate multiple reasoning trajectories (line 2), treating each continuous thought vector as an individual reasoning step. In our configuration, where COCONUT generates $T = 6$ latent vectors per trajectory, we annotate all 6 vectors independently.

For each reasoning step s_i in a trajectory, we generate N candidate completions from that step (line 11) and evaluate their final answers $\{a_j\}_{j=1}^N$ against the ground truth answer a^*. The hard estimation (line 13) of the step-wise rewards is computed as:

$$y_{s_i}^{HE} = \begin{cases} 1 & \exists a_j \in \{a_1, ..., a_N\}, a_j = a^* \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

and the soft estimation (line 14) is computed as:

$$y_{s_i}^{SE} = \frac{\sum_{j=1}^N \mathbb{I}(a_j = a^*)}{N} \tag{3}$$

The hard estimation assigns a positive label if any completion from step s_i leads to the correct answer, indicating that the step preserves the potential for correct reasoning. For trajectory-level evaluation, we compute outcome rewards as: $r^{OUT} = \mathbb{I}[\text{final answer is correct}]$.

4.2 Modeling

A fundamental distinction between continuous and text-based reasoning lies in data transferability across architectures. In text-based reward modeling, data generation and RMs can use different architectures since both operate in a shared textual space. However, continuous thought representations are model-specific, in which only the originating model can interpret its own latent reasoning steps. This restricts us to using COCONUT itself as the backbone for both PRM and ORM.

We augment the COCONUT backbone with task-specific prediction heads: classification heads for PRM hard labels and ORM outcome labels, and a regression head for PRM soft labels. Each head consists of two linear layers with ReLU activation in between. A sigmoid function is applied on all heads to map output values $\in [0, 1]$. For PRM training, we employ joint loss combining cross-entropy for hard estimation and mean squared error for soft estimation:

$$\mathcal{L}_{PRM} = \mathcal{L}_{CE}(y_{s_i}^{HE}, \hat{y}_{s_i}^{HE}) + \mathcal{L}_{MSE}(y_{s_i}^{SE}, \hat{y}_{s_i}^{SE}) \tag{4}$$

For ORM training, we use cross-entropy loss only:

$$\mathcal{L}_{ORM} = \mathcal{L}_{CE}(r^{OUT}, \hat{r}^{OUT}) \tag{5}$$

4.3 Experiments

4.3.1 Experimental Setup

For RM training data construction, we utilize the GSM8k-aug training set, generating M=5 reasoning paths per problem and deduplicating based on final answers (resulting in 1.32 samples per problem on average). We annotate every trajectory step regardless of answer correctness (maintaining consistency with the MATH-Shepherd open-sourced

N	# Unique Answer	# Correct Answers	# Major Incorrect Answers	Pass@N	Confidence	Self-Consistency	PRM-HE	PRM-SE	ORM
1	1.00	0.31	0.69	30.71	30.71	30.71	30.71	30.71	30.71
2	1.36	0.61	1.11	34.57	31.08	30.71	**31.08**	30.78	30.93
4	1.70	1.23	2.14	37.76	30.71	**31.84**	**31.84**	31.61	31.08
8	2.14	2.45	4.11	40.33	30.25	32.15	**32.83**	32.15	31.24
16	2.66	4.92	7.98	43.14	31.01	32.30	**33.51**	33.28	31.84
32	3.18	9.86	15.81	44.96	31.01	31.92	**34.04**	33.21	32.68

Table 1: BoN Performance comparison of different reranking methods on GSM8k. Pass@N shows theoretical upper bounds, while PRM and ORM achieve modest improvements over baselines (Confidence, Self-Consistency) but fall short of the potential indicated by oracle selection. Best results among reranking methods are highlighted in **bold**.

dataset[2] to ensure complete trajectory coverage) and generate $N = 10$ completions for MC estimation during step-wise annotation.

To ensure training stability, we balance positive and negative samples, maintaining a 1:1 ratio. PRM uses step-wise hard estimation labels while ORM uses outcome labels. The resulting training sets contain 238k samples for PRM and 324k samples for ORM. Both models are trained for 10 epochs with peak learning rate 1e-4 (500 warmup steps), batch size 128, on a single A100 GPU.

For evaluation, we employ Best-of-N (BoN) methodology (Stiennon et al., 2022) with $N \in \{1, 2, 4, 8, 16, 32\}$ on the GSM8k test set, retaining all candidates without deduplication to assess genuine ranking effectiveness. We compare RM-based verification against confidence-based reranking using answer probability scores and self-consistency via majority voting (Wang et al., 2023). Beyond accuracy metrics, we analyze the average number of unique answers, correct answers among N candidates, and majority incorrect answers to provide comprehensive evaluation insights.

4.3.2 Experimental Results

As shown in Table 1, Pass@N upper bounds demonstrate substantial potential for inference-time scaling, consistent with our preliminary study (Figure 1). However, model-intrinsic approaches prove inadequate: confidence-based reranking provides no improvement, indicating poor calibration in COCONUT, while self-consistency shows marginal gains but remains limited. The underlying issue becomes apparent when examining answer distributions: across all N values, correct answers consistently fall below dominant incorrect answers, revealing COCONUT's inability to achieve effective scaling through intrinsic capabilities alone.

[2]https://huggingface.co/datasets/peiyi9979/Math-Shepherd

Both PRM and ORM outperform model-intrinsic methods, with PRM-HE achieving the most consistent improvements, reaching 34.04% accuracy at N=32 compared to the 31.92% baseline. While this confirms RMs' potential utility, the improvement magnitude is limited, still far away from the theoretical upper bound (44.96%). These gains pale compared to text-based RM verification, where similar methodologies yield substantially larger improvements (Wang et al., 2024).

Investigation of different score aggregation strategies (min, max, mean, last-step) following Zhang et al. (2025) and a variety of combinations of rerankers shows negligible variation across methods (Table 10 in Appendix B.2), suggesting that core limitations transcend simple scoring mechanisms and necessitate deeper analysis of the reasoning mechanism of COCONUT and RM capabilities. Additional analysis on the effects of hyperparameters on reasoning performance and the role of continuous thoughts are provided in Appendix A.

5 Analysis

The substantial gap between RM reranking performance and theoretical upper bounds motivates a detailed investigation into the root causes of RM ineffectiveness. We hypothesize two potential explanations for this limitation. First, RMs may suffer from fundamental discriminative capacity constraints, failing to reliably distinguish between correct and incorrect reasoning paths or final answers. To examine this possibility, we conduct a systematic evaluation of RM classification performance. Second, the distribution of COCONUT's continuous reasoning trajectories may exhibit weak correlation with answer correctness, preventing RMs from extracting meaningful discriminative features. To investigate this hypothesis, we analyze COCONUT's continuous reasoning from two com-

	Accuracy	Precision	Recall	F1-Score
PRM	62.98	41.60	77.28	54.09
ORM	73.72	39.11	75.76	51.59
	AUROC	**PR-AUC**	**ECE ↓**	
PRM	76.19	61.08	29.33	
ORM	82.22	61.75	23.48	

Table 2: Trajectory-level performance comparison between PRM and ORM. Lower ECE indicates better probabilistic calibration.

plementary perspectives: geometric properties of thought representations and trajectory dynamics.

5.1 Curation of Evaluation Dataset

For subsequent experiments, we construct a dedicated test set for systematic evaluation based on the GSM8k test set, employing the same annotation methodology used for training data construction (as detailed in Algorithm 1). To ensure high-quality annotations, we increase the number of generated samples per problem to $M = 10$ (retaining an average of 2.29 answers per problem after deduplication) and expand the MC estimation candidate size to $N = 20$ for enhanced label reliability. The resulting evaluation dataset contains 3,014 samples with 18.48% correct answers and 28.21% correct reasoning steps. We deliberately preserve the natural distribution without rebalancing to accurately reflect RM performance across the full spectrum of correct and incorrect reasoning patterns.

5.2 Classification Performance of RMs

We evaluate the classification performance of both PRM (hard estimation prediction) and ORM using our constructed test set, employing a threshold of 0.5 to distinguish between correct and incorrect steps or solutions for both models.

As shown in Table 2, both PRM and ORM exhibit weak performance when treated as binary classifiers, with low F1-scores (PRM: 54.09%, ORM: 51.59%). The confusion matrices in Figure 4 in Appendix B.1 further reveal systematic failure modes: PRM produces a large number of false positives (5,535 cases, 30.6%) relative to true positives (3,943 cases, 21.8%), resulting in poor precision (41.60%) despite relatively high recall (77.28%), indicating that incorrect reasoning steps are frequently assigned high confidence. Although ORM achieves higher overall accuracy (73.72%), it similarly suffers from low precision (39.11%), suggesting that misclassification remains prevalent across both RMs. Beyond threshold-dependent metrics, AUROC and PR-AUC provide a threshold-free view of discriminative capability, where ORM demonstrates stronger ranking performance than PRM; however, the modest PR-AUC values indicate limited effectiveness under class imbalance. Consistent with these observations, the calibration curves in Figure 5 in Appendix B.1 show substantial overconfidence for both models, with predicted confidence systematically exceeding empirical accuracy. While ORM exhibits lower ECE (23.48 vs. 29.33), both RMs remain poorly calibrated, indicating that their score magnitudes are unreliable as probability estimates.

Taken together, these findings explain the limited reranking gains observed in our experiments. Both models suffer from poor calibration and low precision, fundamentally constraining their ability to reliably identify and prioritize correct reasoning paths. This suggests that deeper structural challenges inherent to continuous space reward modeling cannot be addressed through simple methodological adjustments.

5.3 Geometric Properties of Thought Representation

In this analysis, we treat thought vectors as independent representations and examine their geometric properties using two key metrics to understand their spatial distribution characteristics.

We first consider isotropy, which measures the uniformity of variance across all dimensions, revealing how thought vectors distribute and utilize dimensions in high-dimensional space. We employ IsoScore* proposed by Rudman and Eickhoff (2024) due to its superior properties, including mean agnosticism, rotation invariance, and global stability. IsoScore* operates by computing the covariance matrix eigenvalues $\Lambda = \{\lambda_1, ..., \lambda_d\}$ of input thought vectors $\mathbf{s}$, and normalizing them as $\hat{\Lambda} = \sqrt{d} \cdot \Lambda / ||\Lambda||_2$, where d represents the dimensionality of thought vectors. The final score is calculated as:

$$\begin{aligned} \delta(\Lambda) &= \frac{||\hat{\Lambda} - \mathbf{1}||}{\sqrt{2(d - \sqrt{d})}}, \quad \mathbf{1} = (1, ..., 1)^T \in \mathbb{R}^d \\ \phi(\Lambda) &= \frac{(d - \delta(\Lambda)^2 (d - \sqrt{d}))^2}{d^2} \\ \text{IsoScore}^\star &= \frac{d \cdot \phi(\Lambda) - 1}{d - 1} \end{aligned} \tag{6}$$

A value of 1 indicates perfect isotropy, while 0 indicates maximal anisotropy.

Metrics	IsoScore⋆		Hoyer	
Groups	Correct	Incorrect	Correct	Incorrect
Entire set	0.0134	0.013	0.21 ± 0.01	0.22 ± 0.01
PRM+	0.0137	0.0131	0.21 ± 0.01	0.22 ± 0.01
PRM-	0.0126	0.0132	0.22 ± 0.01	0.21 ± 0.01

Table 3: Geometric properties with Hoyer and IsoScore⋆. Results are grouped with the entire set, PRM+ (samples correctly predicted by the PRM), PRM- (samples incorrectly predicted by the PRM). Values are computed within subsets of thought vectors annotated with their correctness based on the hard estimation.

We also examine sparsity using the Hoyer (Hurley and Rickard, 2009) metric to assess dimensional activation patterns:

$$Hoyer(s_i) = \frac{\sqrt{d} - ||s_i||_1/||s_i||_2}{\sqrt{d-1}} \quad (7)$$

Higher Hoyer values indicate sparser activations where fewer dimensions carry primary information, corresponding to more focused reasoning representations, while lower values suggest distributed activation patterns.

We evaluate both metrics across three sample groups: all thoughts in the test set (noted as Entire Set), thoughts correctly predicted by PRM (where PRM prediction matches ground truth, noted as PRM+), and thoughts incorrectly predicted by PRM (where PRM prediction differs from ground truth, noted as PRM-). Within each group, we separately analyze thoughts labeled as "correct" and "incorrect."

Table 3 reveals several critical findings: (1) Thought representations exhibit low isotropy and relative sparsity, confirming that reasoning operates within limited dimensional subspaces. (2) Most importantly, across all measurement groups, the differences between correct and incorrect thoughts are negligible, indicating that geometric properties (present in COCONUT thought vectors) alone cannot distinguish reasoning step correctness. This fundamental lack of geometric separability explains why PRM fails to effectively discriminate through these features. The t-SNE visualization in Figure 2 further confirms this observation, showing that correct and incorrect thoughts are completely intermixed in the representation space [3]. In Appendix A.1, we further conduct case-wise analysis on the separability of the last-step thought vector and obtain consistent findings.

[3]The two distinct clusters are due to 2 vectors per reasoning step ($c = 2$), where the model differentiates between vector positions within each step. This clustering is unrelated to reasoning correctness.

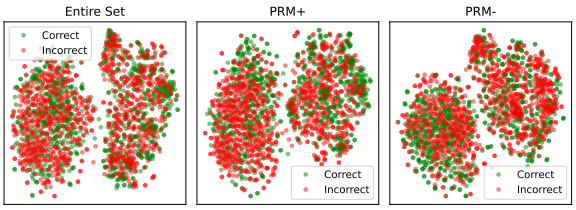

Figure 2: t-SNE of the sampled latent thoughts labeled with correct/incorrect for the entire set, PRM+ and PRM- groups.

Metric	Correct	Incorrect	p-value	Cohen's d
Entire Set				
Compactness	19.81 ± 2.53	19.39 ± 2.48	**0.023***	0.17
Straightness	0.22 ± 0.04	0.21 ± 0.04	0.637	0.04
Curvature	9.32 ± 0.52	9.38 ± 0.53	0.161	−0.10
Local Smoothness	0.43 ± 0.11	0.44 ± 0.11	0.074	−0.13
ORM+				
Compactness	19.91 ± 2.26	19.56 ± 2.42	0.295	0.15
Straightness	0.22 ± 0.04	0.21 ± 0.05	0.553	0.09
Local Smoothness	0.42 ± 0.10	0.45 ± 0.11	0.089	−0.24
Curvature	9.30 ± 0.42	9.40 ± 0.50	0.169	−0.20
PRM+				
Compactness	20.72 ± 1.97	18.55 ± 1.83	**0.022***	1.14
Straightness	0.22 ± 0.03	0.23 ± 0.03	0.490	−0.32
Local Smoothness	0.39 ± 0.09	0.48 ± 0.10	**0.049***	−0.97
Curvature	8.87 ± 0.59	8.90 ± 0.52	0.884	−0.07

Table 4: Comparison of geometric metrics between correct and incorrect reasoning chains on the entire set, ORM+ subset (ORM correctly predicted samples), and PRM+ subset (PRM correctly predicted samples). Bold p-values indicate statistical significance ($p < 0.05$).

5.4 Trajectory Dynamics

In this analysis, we examine COCONUT's reasoning process from a trajectory perspective to investigate whether reasoning correctness is reflected in trajectory dynamics. Specifically, we analyze the geometric properties of continuous reasoning trajectories using four complementary metrics. We first examine **compactness**, which quantifies how tightly trajectory points cluster around their centroid:

$$\bar{\mathbf{s}} = \frac{1}{T}\sum_{i=1}^{T}\mathbf{s}_i \quad R_g^2 = \frac{1}{T}\sum_{i=1}^{T}\|\mathbf{s}_i - \bar{\mathbf{s}}\|_2^2$$
$$\text{compactness} = \sqrt{R_g^2} \quad (8)$$

where T is the number of reasoning steps, $\mathbf{s}_i$ represents the i-th thought vector, and $\bar{\mathbf{s}}$ is the trajectory centroid. Smaller compactness values indicate trajectories that remain within a concentrated region of reasoning space. Next, we measure **curvature** (Hosseini and Fedorenko, 2023), capturing

the total bending energy along the reasoning path:

$$\boldsymbol{\Delta}_i = \mathbf{s}_{i+1} - \mathbf{s}_i \tag{9}$$

$$\theta_i = \arccos\left(\frac{\boldsymbol{\Delta}_{i-1} \cdot \boldsymbol{\Delta}_i}{\|\boldsymbol{\Delta}_{i-1}\|_2 \|\boldsymbol{\Delta}_i\|_2}\right), i \geq 2$$

$$\text{curvature} = \sum_{i=2}^{T-1} \theta_i \tag{10}$$

where $\boldsymbol{\Delta}_i$ represents the displacement vector between consecutive thoughts, and θ_i is the angle between successive displacement vectors. We assess **local smoothness** by quantifying coherence between consecutive reasoning steps using cosine similarity:

$$\text{local\_smoothness} = \frac{1}{T-1} \sum_{i=1}^{T-1} \frac{\mathbf{s}_i \cdot \mathbf{s}_{i+1}}{\|\mathbf{s}_i\|_2 \|\mathbf{s}_{i+1}\|_2} \tag{11}$$

Finally, we evaluate **straightness**, measuring global path efficiency by comparing net displacement to total path length:

$$L = \sum_{i=1}^{T-1} \|\mathbf{s}_{i+1} - \mathbf{s}_i\|_2 \quad D = \|\mathbf{s}_T - \mathbf{s}_1\|_2$$

$$\text{straightness} = \frac{D}{L} \in [0, 1] \tag{12}$$

where L is the total path length and D is the net displacement from start to end. Values approaching 1 indicate efficient, direct reasoning paths, while lower values suggest circuitous or exploratory trajectories.

Similar to our geometric analysis, we analyze three sample groups: all trajectories in the test set, trajectories correctly predicted by ORM, and trajectories where PRM correctly predicts all steps. For each group, we calculate metrics for trajectories labeled as "correct" or "incorrect" and perform t-tests. As shown in Table 4, most metrics show minimal differences between correct and incorrect trajectories. While compactness shows statistical significance in the entire set ($p = 0.023$) and PRM+ subset ($p = 0.022$), the effect sizes remain small (Cohen's d=0.17 and 1.14 respectively). The most notable finding occurs in the PRM+ subset, where correct trajectories exhibit higher compactness and lower local smoothness, potentially suggesting that correct reasoning paths are more spatially concentrated yet less smooth between consecutive steps. Overall, the results demonstrate that correct and incorrect reasoning trajectories exhibit no substantial differences in trajectory dynamics, further limiting the possibility of RMs to leverage these characteristics for effective discrimination.

	$\lvert\mu^+ - \mu^-\rvert$		p-value		Cohen's d	
Metrics	**Base**	**Tuned**	**Base**	**Tuned**	**Base**	**Tuned**
Cpt	0.75	**2.50**	1.1e-04	**3.8e-09**	**1.08**	1.05
Str	0.01	**0.03**	4.4e-01	**4.3e-02**	0.99	**1.02**
Lsm	0.03	**0.08**	2.9e-01	**8.3e-03**	0.98	**1.08**
Cur	0.15	**0.27**	1.6e-02	**5.5e-03**	0.91	**1.09**

Table 5: Trajectory dynamics analysis ($N = 32$) before and after post-training, including the absolute mean gap $|\mu^+ - \mu^-|$, two-sided t-test p-values, and Cohen's d effect sizes for correct versus incorrect trajectories.

	Pass@N		PRM-HE-last		ORM	
N	**Baseline**	**Tuned**	**Baseline**	**Tuned**	**Baseline**	**Tuned**
1	30.71	**31.24**	30.71	31.24	30.71	31.24
2	34.57	**35.41**	31.08	**31.92**	30.93	**31.08**
4	37.76	**40.03**	31.84	**33.43**	31.08	**31.54**
8	40.33	**43.14**	32.83	**33.81**	31.24	**32.75**
16	43.14	**45.56**	33.51	**33.95**	31.84	**32.68**
32	44.96	**48.22**	34.04	**34.57**	**32.68**	32.37

Table 6: Inference-time scaling performance before and after post-training, including Pass@N and reranked accuracy using PRM-HE-last and ORM, evaluated at different sampling budgets N.

6 Post-training with Geometric Regularization

Our analyses suggest that the bottleneck of COCONUT's inference-time scaling arises from limited separability of latent thoughts associated with different answers, hindering effective discrimination by reward models or rerankers. To verify whether this limitation can be alleviated, as discussed in Appendix §C, we propose to introduce a lightweight post-training intervention that injects explicit geometric inductive bias inspired by contrastive (van den Oord et al., 2019) learning and RLVR-style (DeepSeek-AI et al., 2025) training.

6.1 Training Objective

During post-training, we perform dropout-based rollouts to sample N trajectories per prompt and verify their correctness using ground-truth answers. Let $g(\tau) : \mathbb{R}^{T \times D} \to \mathbb{R}$ denote a trajectory-level geometric metric (e.g., compactness) computed on a T-step reasoning trajectory $\tau = \{\mathbf{s}_1, ..., \mathbf{s}_T\}$. For each prompt x_i, we obtain a set of N trajectories $\{\tau_{i,j}\}_{j=1}^N$ and their correctness labels $y_{i,j} \in \{0, 1\}$ by verifying the generated answers against the ground-truth. Because different metrics have heterogeneous scales, we first apply *prompt-wise z-*

score normalization over the N trajectories:

$$\tilde{g}(\tau_{i,j}) \;=\; \frac{g(\tau_{i,j}) - \mu_{g,i}}{\sigma_{g,i} + \epsilon} \tag{13}$$

where $\epsilon = 1e-12$. We then compute the mean normalized metric value for correct and incorrect trajectories:

$$\mu_{g,i}^{(\pm)} = \frac{\sum_{j=1}^{N} \mathbb{I}[y_{i,j} = k]\, \tilde{g}(\tau_{i,j})}{\sum_{j=1}^{N} \mathbb{I}[y_{i,j} = k] + \epsilon}, \quad k \in \{0, 1\} \tag{14}$$

where $k = 1$ and $k = 0$ correspond to $\mu_{g,i}^{+}$ and $\mu_{g,i}^{-}$, respectively. We then define the per-prompt separation as $\Delta_{g,i} = \mu_{g,i}^{+} - \mu_{g,i}^{-}$. We only apply the geometric regularizer to prompts that have *both* correct and incorrect trajectories (i.e., excluding rollouts that are all-correct or all-incorrect). The geometric regularization term for metric g is

$$\mathcal{L}_{\text{geo}}(g) \;=\; \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \max\Big(0,\; \gamma - \big|\Delta_{g,i}\big|\Big), \tag{15}$$

where $\mathcal{I}$ denotes the set of valid prompts in the minibatch and we fix $\gamma = 1.0$ for all four metrics in our experiments. Let $\mathcal{G}$ be the set of the four trajectory-dynamics metrics (compactness, straightness, local smoothness, and curvature) used in this work. The final geometry loss is the average across metrics:

$$\mathcal{L}_{\text{geo}} \;=\; \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \mathcal{L}_{\text{geo}}(g). \tag{16}$$

To prevent the model from drifting away from its original answer-generation behavior, we additionally apply a standard cross-entropy loss on *correct* trajectories only, denoted as $\mathcal{L}_{\text{ce}}^{+}$. The overall post-training objective is

$$\mathcal{L} \;=\; \mathcal{L}_{\text{ce}}^{+} + \lambda \mathcal{L}_{\text{geo}} \tag{17}$$

where $\lambda = 1.0$ in our experiment.

6.2 Experimental Setup

We perform the post-training on a random subset of 5000 examples sampled from the GSM8k training split. The training runs for a single epoch and updates only the final Transformer block of the COCONUT backbone (all other parameters are frozen). We use AdamW with batch size 4 prompts (each prompt produces $N = 32$ trajectories after rollout, yielding $4N$ trajectory instances per minibatch), a warmup of 100 steps, and a peak learning rate of 5×10^{-5}.

Evaluation Protocol For separability analysis (Table 5), we rollout $N = 32$ trajectories per prompt on the GSM8k test set using the post-trained model. We remove prompts whose rollouts are all-correct or all-incorrect, sort the remaining prompts by the balance of correctness ratio, and select the top 100 prompts as analysis candidates. On this subset, we compute the absolute mean gaps $|\mu_g^{+} - \mu_g^{-}|$ and perform two-sided t-tests. Note that, for this *evaluation* step, we report μ_g^{+} and μ_g^{-} computed from the *raw* (unnormalized) metric values to reflect the true magnitude of separation. For inference-time scaling evaluation, we follow the configuration described in §4.3.1, replacing the generator with the post-trained model while keeping all other settings unchanged.

6.3 Experimental Result

On GSM8k, post-training yields significantly larger inter-group mean gaps across all metrics (Table 5) and improved Pass@N performance (Table 6). Although we do not re-run the full reward model training pipeline, PRM and ORM still achieve modest gains over the baseline, suggesting that the post-training procedure preserves the original latent structure and remains compatible with existing RMs; we view this as a proof-of-concept and leave more effective metric selection and training objectives for further improving inference-time scaling to future work.

7 Conclusion

In this work, we study inference-time scaling for continuous reasoning models via multiple sample generation and reward-model reranking using COCONUT. Although preliminary results show strong Pass@N potential, established text-based reward modeling methods transfer poorly to continuous reasoning. Our analysis shows that continuous thoughts lack the geometric separability needed to distinguish correct from incorrect reasoning, with only minimal structural differences between them. We further show that a lightweight post-training intervention with geometric inductive bias can partially mitigate this issue. Overall, our findings suggest that effective inference-time scaling in continuous reasoning requires training frameworks that explicitly encourage geometric differentiation in continuous thought space.

Limitation

This work represents an early study of inference-time scaling for continuous reasoning using COCONUT as the primary testbed, and thus has several limitations. First, our empirical analysis is conducted on a single reasoning benchmark (GSM8k) with a fixed continuous reasoning configuration, and the observed limitations of reward-model reranking may not directly generalize to other tasks or continuous reasoning architectures. Second, the training and evaluation of PRM/ORM rely on automatic annotation and sampling-based procedures, which introduce sensitivity to sampling hyperparameters and may affect the stability of reward signals. Finally, our geometric analysis and post-training intervention focus on a small set of trajectory-level metrics and a lightweight regularization design; exploring richer geometric representations and more principled training objectives remains an open direction for future work.

References

Jeffrey Cheng and Benjamin Van Durme. 2024. Compressed chain of thought: Efficient reasoning through dense representations. *Preprint*, arXiv:2412.13171.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, and Junxiao Song et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2019. Universal transformers. *Preprint*, arXiv:1807.03819.

Yuntian Deng, Yejin Choi, and Stuart Shieber. 2024. From explicit cot to implicit cot: Learning to internalize cot step by step. *Preprint*, arXiv:2405.14838.

Halil Alperen Gozeten, M. Emrullah Ildiz, Xuechen Zhang, Hrayr Harutyunyan, Ankit Singh Rawat, and Samet Oymak. 2025. Continuous chain of thought enables parallel exploration and reasoning. *Preprint*, arXiv:2505.23648.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *Preprint*, arXiv:2412.06769.

Eghbal A. Hosseini and Evelina Fedorenko. 2023. Large language models implicitly learn to straighten neural sentence trajectories to construct a predictive representation of natural language. *Preprint*, arXiv:2311.04930.

Niall P. Hurley and Scott T. Rickard. 2009. Comparing measures of sparsity. *Preprint*, arXiv:0811.4706.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhiwei Li, Bao-Long Bi, Ling-Rui Mei, Junfeng Fang, Xiao Liang, Zhijiang Guo, and 2 others. 2025. From system 1 to system 2: A survey of reasoning large language models. *Preprint*, arXiv:2502.17419.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

Amirkeivan Mohtashami, Matteo Pagliardini, and Martin Jaggi. 2024. Cotformer: A chain-of-thought driven architecture with budget-adaptive computation cost at inference. *Preprint*, arXiv:2310.10845.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, and Aiden Low et al. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

William Rudman and Carsten Eickhoff. 2024. Stable anisotropic regularization. *Preprint*, arXiv:2305.19358.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. 2025. Codi: Compressing chain-of-thought into continuous space via self-distillation. *Preprint*, arXiv:2502.21074.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2022. Learning to summarize from human feedback. *Preprint*, arXiv:2009.01325.

DiJia Su, Hanlin Zhu, Yingchen Xu, Jiantao Jiao, Yuandong Tian, and Qinqing Zheng. 2025. Token assorted: Mixing latent and text tokens for

improved language model reasoning. *Preprint*, arXiv:2502.03275.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *Preprint*, arXiv:2503.16419.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *Preprint*, arXiv:2211.14275.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding. *Preprint*, arXiv:1807.03748.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *Preprint*, arXiv:2312.08935.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. The lessons of developing process reward models in mathematical reasoning. *Preprint*, arXiv:2501.07301.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. *Preprint*, arXiv:2205.10625.

Rui-Jie Zhu, Tianhao Peng, Tianhao Cheng, Xingwei Qu, Jinfa Huang, Dawei Zhu, Hao Wang, Kaiwen Xue, Xuanliang Zhang, Yong Shan, Tianle Cai, Taylor Kergan, Assel Kembay, Andrew Smith, Chenghua Lin, Binh Nguyen, Yuqi Pan, Yuhong Chou, Zefan Cai, and 14 others. 2025. A survey on latent reasoning. *Preprint*, arXiv:2507.06203.

ID	% Correct	Iso+	Iso-	Hoyer+	Hoyer-	p-value	Cohen's d
1	53.9	0.014	0.014	0.370	0.359	0.075	0.315
2	52.3	0.011	0.010	0.480	0.496	**0.044**	0.359
3	47.7	0.007	0.004	0.432	0.429	0.701	0.068
4	46.1	0.010	0.008	0.437	0.447	0.268	0.197
5	42.1	0.019	0.021	0.365	0.357	0.156	0.251
6	41.4	0.016	0.015	0.455	0.467	0.149	0.261
7	41.4	0.019	0.016	0.466	0.460	0.411	0.150
8	40.6	0.019	0.018	0.398	0.401	0.533	0.108

Table 7: Geometric analysis including Isotropy and Sparsity (Hoyer with t-test) on the 8 cases, applied within the correct (+) and incorrect (group) of $N = 128$ reasoning trajectories. The correct percent is also presented for each case.

p	# Unique Answer	# Correct Answers	# Major Incorrect Answers	Pass@N	Confidence	Self Consistency	PRM-HE
0.01	1.69	10.8	18.65	39.5	33.88	34.12	35.1
0.05	2.55	10.01	16.66	44.03	32.29	33.28	34.87
0.1	3.18	9.86	15.81	44.96	31.01	31.92	34.04
0.3	4.45	6.14	16.90	38.67	18.80	21.15	21.30
0.5	3.56	2.79	21.02	21.68	8.04	9.55	7.58
0.7	4.23	2.45	20.09	21.23	6.75	7.88	7.35
0.9	5.95	2.16	17.38	22.90	6.52	7.43	7.51

Table 8: Sensitivity analysis of dropout-based sampling under different dropout probabilities p with $N = 32$ samples.

Appendix

A Additional Analysis on COCONUT

In this section, we conduct additional analyses on the reasoning process of COCONUT. We first perform a small-scale case study in which dropout is applied only at the final step of the model's reasoning. Our results confirm again that these thought representations remain difficult to distinguish in the embedding space. We then investigate thought sampling under different dropout rates and find that the optimal dropout rate should remain consistent with the training configuration. Finally, we experiment with injecting Gaussian noise into the thought representations and examine the resulting model behavior. Interestingly, even when the thought representations are entirely replaced by noise, the model is still able to perform a non-trivial level of reasoning. This observation suggests that, for certain problems, the model may not strongly rely on continuous thoughts for reasoning, and that such representations may instead act as placeholders rather than encoding essential reasoning content.

A.1 Case Study

To reduce dataset-level bias caused by prompt-specific fluctuations in geometric measurements, we conduct a controlled case study in which dropout sampling is applied only at the final latent reasoning step. We run COCONUT on the GSM8K test set with $N = 128$ to obtain sufficient reasoning trajectories. Prompts with all-correct or all-incorrect samples are excluded, and the remaining prompts are ranked by their degree of balance of answer correctness, from which the eight most balanced cases (including $\sim 50\%$ correct answers) are selected.

Figure 3 visualizes the final-step latent representations for these cases, annotated with answer correctness. Correct and incorrect answers are thoroughly mixed, with no observable clustering among the types of answers. Table 7 further reports geometric metrics (IsoStar and Hoyer) and hypothesis tests comparing correct and incorrect groups. Except for one case with a marginally significant Hoyer p-value but negligible effect size, no meaningful differences are observed. These findings are fully consistent with the dataset-level analysis presented in Section 5.3, providing further evidence that COCONUT's latent thoughts are not easily separable in the hidden representation space.

A.2 Sensitivity on Dropout Rate

We conduct an additional sensitivity analysis to examine how the dropout rate affects dropout-based sampling and reranking performance. In this experiment, we fix the sampling size to $N = 32$ and vary the dropout probability $p \in \{0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9\}$, while keeping all other inference and reranking configurations unchanged. The setting $p = 0.1$ corresponds to the dropout rate used during training. For each value of p, we report answer-level statistics, including the number of unique answers, correct answers, and dominant incorrect answers, as well as overall Pass@N and reranking performance using confidence-based, self-consistency, and PRM hard-estimation (PRM-HE) reranking results.

As shown in Table 8, when $p < 0.1$, sampled trajectories may not be diverse enough, leading to inferior Pass@N, although this makes the proportion of correct answers larger and eases the reranking. Increasing the dropout rate ($p > 0.1$) leads to a consistent degradation in both Pass@N and reranking performance, accompanied by a sharp decrease in the number of correct answers and an increase in dominant incorrect. While moderate increases in p initially raise answer diversity, this diversity is largely unproductive and quickly overwhelms

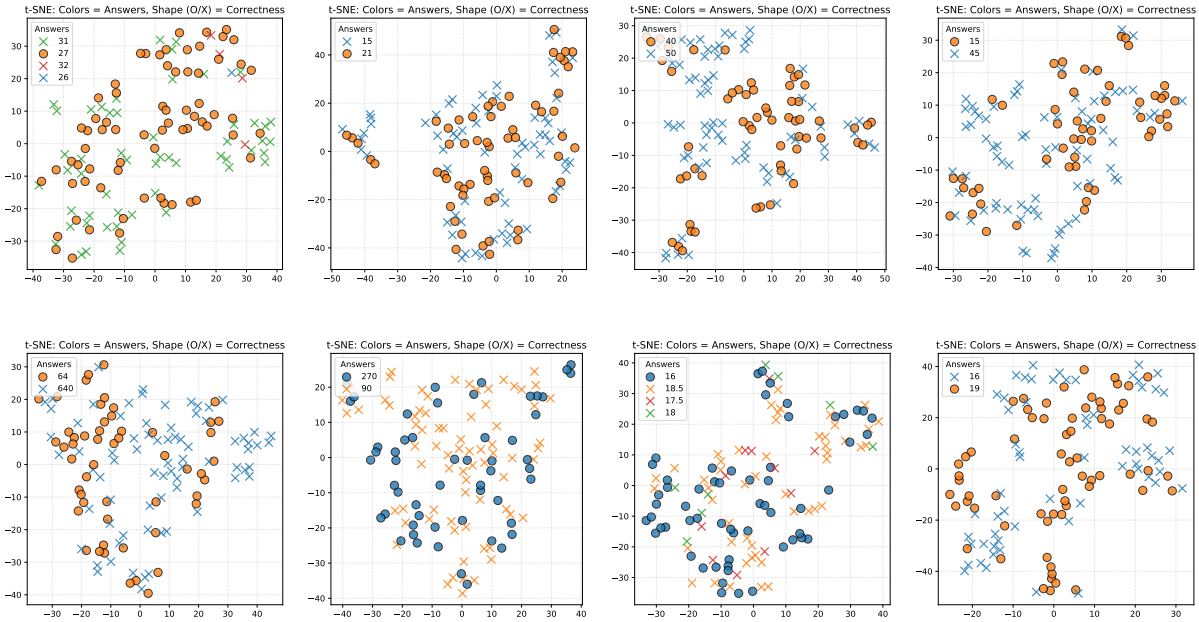

Figure 3: t-SNE visualization on the last reasoning step of the 8 cases, each representation is shaped and colored (○ for correct, × for incorrect) with the correctness and the value of the corresponding answer.

Noise Ratio	# Unique Answer	Pass@5	# Correct Answer	% Majority Answer Unchange
0.0	1.86	39.20	1.55	100.00
0.2	1.92	38.67	1.50	76.35
0.4	2.22	34.80	1.24	67.32
0.6	2.56	20.32	0.55	44.73
0.8	2.62	15.62	0.37	49.43
1.0	2.49	12.59	0.32	53.83

Table 9: Noise perturbation results in different noise levels.

correct trajectories. For $p > 0.5$, performance stabilizes at a uniformly low level, suggesting that overly aggressive dropout disrupts the latent reasoning process and limits effective exploration during sampling.

A.3 Reasoning under Perturbation

Given the lack of significant geometric differences between correct and incorrect thought vectors, we investigate whether COCONUT's reasoning paths contain necessary semantic information or merely serve as positional placeholders. We design a perturbation experiment injecting Gaussian noise into latent thoughts at varying intensities (i.e. $\text{ratio} \times \text{noise} + (1 - \text{ratio}) \times \text{thought}$) while observing Pass@5 (with sampling size $N = 5$) performance on GSM8k. Metrics such as the average number of unique/correct answers among the 5 candidates, and the percentage of problems where the majority answer remains unchanged compared to the previous noise level are also computed, serving as a measurement of answer stability.

As shown in Table 9, COCONUT demonstrates robustness at low noise ratios (0.0-0.2), with minimal degradation and 76% of majority answers unchanged. This aligns with our findings of high anisotropy (i.e., low isotropy) and sparsity—noise primarily affects irrelevant dimensions while preserving critical reasoning dimensions.

Performance degradation becomes pronounced at higher noise ratios, but strikingly, even with complete noise corruption (ratio=1.0), Pass@5 remains non-zero at 12.59%. This suggests COCONUT's reasoning does not exclusively depend on latent thoughts, for many problems, the model can generate correct answers independently of continuous reasoning. This raises questions about continuous thoughts' actual contribution.

B Additional details on PRM and ORM

B.1 Details of Classification Performance

B.2 Details of Multiple Reranker Combination

C Discussion

Through systematic analysis of reward model performance and geometric properties, we uncover fundamental challenges in reward modeling for

PRM Hard Estimation												
N	**last**	**min**	**max**	**mean**	**SC+last**	**SC+min**	**SC+max**	**SC+mean**	**Conf+last**	**Conf+min**	**Conf+max**	**Conf+mean**
1	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71
2	31.08	31.08	30.25	31.31	31.08	31.08	30.25	**31.31**	31.08	31.08	30.25	**31.31**
4	31.84	32.22	30.63	**32.45**	31.31	31.46	31.24	31.69	31.84	32.22	30.63	**32.45**
8	32.83	**32.98**	30.86	32.07	32.15	32.15	31.77	32.07	32.83	32.90	30.86	32.07
16	**33.51**	33.13	31.46	32.75	32.45	32.52	32.37	32.45	**33.51**	33.13	31.46	32.75
32	34.04	34.12	32.60	**34.27**	32.52	32.45	31.99	32.45	34.04	34.12	32.60	**34.27**
PRM Soft Estimation												
N	**last**	**min**	**max**	**mean**	**SC+last**	**SC+min**	**SC+max**	**SC+mean**	**Conf+last**	**Conf+min**	**Conf+max**	**Conf+mean**
1	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71
2	30.78	**30.86**	30.78	30.78	30.78	30.86	30.78	30.78	30.78	**30.86**	30.78	30.78
4	31.61	31.46	31.92	**32.22**	31.31	31.39	31.39	31.54	31.61	31.46	31.92	**32.22**
8	**32.15**	30.78	31.31	31.69	**32.15**	31.92	31.92	31.99	**32.15**	30.78	31.31	31.69
16	**33.28**	32.60	32.30	32.30	32.37	32.37	32.30	32.52	**33.28**	32.60	32.30	32.30
32	**33.21**	32.15	31.77	32.90	32.60	32.45	32.30	32.22	33.21	32.15	31.77	32.90
	ORM + SC/Conf		**PRM (HE + SE)**				**PRM (HE+SE) + ORM**				**Other**	
N	**SC+ORM**	**Conf+ORM**	**last**	**min**	**max**	**mean**	**last**	**min**	**max**	**mean**	**SC+Conf**	**RMs+SC+Conf**
1	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71	30.71
2	30.93	30.93	31.08	30.78	30.93	31.08	31.61	31.39	**31.77**	31.08	29.95	31.61
4	31.24	31.08	31.61	31.69	31.84	**32.60**	32.52	32.22	31.92	32.37	31.61	32.45
8	32.22	31.24	32.75	31.31	32.07	32.37	32.37	32.30	32.45	**32.98**	31.84	32.07
16	32.90	31.84	33.51	31.69	32.90	32.45	**33.28**	33.06	32.22	33.06	32.37	32.98
32	32.60	32.68	**34.19**	33.06	**34.19**	33.97	32.60	32.37	32.22	32.90	31.99	33.43

Table 10: Comparison of reranking performance under different combinations of reward-based and heuristic rerankers. The first block reports results dominated by PRM hard estimation (HE), evaluating four aggregation strategies over sampled trajectories (*last*, *min*, *max*, *mean*), as well as their combinations with self-consistency (SC, normalized by sampling size) and confidence-based reranking (Conf). The second block presents corresponding results using PRM soft estimation (SE). The third block evaluates combinations of ORM with SC and Conf. The fourth block combines PRM hard and soft estimation (HE+SE). The fifth block further integrates PRM (HE+SE) with ORM. The final block reports results using SC and Conf alone, as well as the full combination of PRM, ORM, SC, and Conf (with PRM using the *last* aggregation). All rerankers are combined linearly with equal weights (1.0), and all component scores are normalized to the range $[0, 1]$.

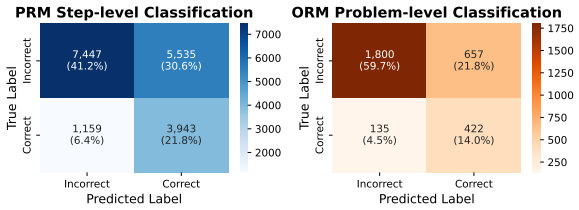

Figure 4: Confusion matrix of PRM and ORM.

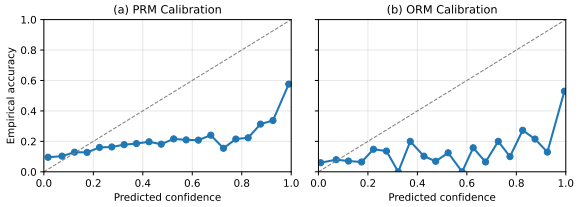

Figure 5: Calibration plot of PRM and ORM.

continuous thought models. Our findings reveal that continuous thoughts cluster within concentrated high-dimensional space without sufficient semantic differentiation. This prevents effective discrimination between correct and incorrect reasoning trajectories.

We attribute this clustering to the absence of inductive biases or geometric structural constraints during COCONUT's training, where supervision applies only to text tokens while latent thoughts are generated without explicit guidance. This results in homogeneous thought vectors that learn shared characteristics without encoding distinctive semantic properties, confounding reward model training.

Our analysis suggests that introducing targeted inductive biases during training could potentially address these limitations. Promising directions include encouraging higher isotropy in thought representations, promoting trajectory diversity through varying geometric patterns, and incorporating contrastive learning to teach discrimination between correct and incorrect reasoning patterns. By establishing clearer geometric structure in continuous thought space, these approaches could enable effective reward modeling and unlock inference-time scaling potential.