Improving Latent Reasoning in LLMs via Soft Concept Mixing

Anonymous ACL submission

Abstract

Unlike human reasoning in abstract conceptual spaces, large language models (LLMs) typically reason by generating discrete tokens at each step, then discard the rich semantic information contained in the output distribution. Recent work such as Soft Thinking has shown that LLMs' latent reasoning via soft concepts is a promising direction, but LLMs are trained on discrete tokens, leaving the continuous concept space underutilized during training. To address this, we propose Soft Concept Mixing (SCM), a soft concept aware training scheme that directly exposes the model to soft representations during training. Specifically, SCM constructs a soft concept vector by forming a probability-weighted average of embeddings. Then, this vector is mixed into the model's hidden states, which embody rich contextual information. Finally, the entire latent reasoning process is optimized with Reinforcement Learning (RL). Experiments on six reasoning benchmarks demonstrate that SCM improves the reasoning performance of LLMs, and simultaneously maintains a stable training dynamic.

1 Introduction

Large language models (LLMs) excel in complex reasoning tasks like mathematics, commonsense, and code. Chain-of-Thought (CoT) prompting (Wei et al., 2022; Lobo et al., 2025) and its related works (Guo et al., 2025; Zhang et al., 2022) substantially improve LLM performance by decomposing problems step by step through the generation of intermediate reasoning traces.

However, unlike human cognition which operates in a high-dimensional abstract space to weigh multiple possibilities(Quiroga et al., 2005; Fedorenko and Varley, 2016), standard CoT reasoning constrains the reasoning process strictly to sequences of discrete tokens(Yao et al., 2023). This constraint may introduce a significant risk of error propagation, where a minor deviation in early reasoning steps—often stemming from sampling randomness or greedy decoding limitations—cascades into an incorrect final conclusion (Turpin et al., 2023). Crucially, the underlying probability distribution often retains the correct reasoning direction within the top-ranked logits, yet this rich semantic information regarding alternative hypotheses is entirely discarded during the discretization process.

Recent studies have explored latent space reasoning (Deng et al., 2025; Xu et al., 2025b; Cheng and Van Durme, 2024; Tack et al., 2025; Yang et al., 2024). For instance, Coconut (Hao et al., 2024) uses the last hidden state of the model as the next-step input embedding, thereby achieving a form of "continuous thought" within the latent space. However, its training paradigm relies on a complex, multi-stage training process that requires a large corpus of CoT trajectories for supervision. This approach is not only computationally intensive but also potentially degrades the general capabilities acquired during pre-training. In contrast, a Reinforcement Learning (RL) framework does not require CoT trajectories, enabling the model to explore alternative reasoning paths through outcome-driven optimization.

Another line of work, Soft Thinking (Zhang et al., 2025), leverages the full output probability distribution to construct a concept token, allowing reasoning to proceed in a high-dimensional semantic space. However, as a method applied only at inference time, it introduces a fundamental gap between its continuous reasoning process and the model's training on discrete tokens. Since the model is never exposed to such continuous representations during its training phase, it struggles to utilize these soft concepts, causing performance instability. This requires the development of a novel framework that empowers the model to effectively explore and internalize continuous concepts, weaving them directly into the fabric of its own rich,

contextual understanding during training.

To achieve this, we propose **S**oft **C**oncept **M**ixing (**SCM**), a lightweight framework that integrates soft concept vectors directly into the language model's training phase. At each reasoning step, SCM not only samples discrete tokens but also leverages the model's full output probability distribution to generate a soft concept vector via probability-weighted average of embeddings. This vector is mixed into the model's hidden states, preserving the distribution's rich latent semantics. SCM additionally introduces a continuous context conditioning pathway between successive steps. This allows gradients from subsequent steps to influence the probability distribution at step t, complementing the RL exploration with continuous context optimization. To optimize the complex policy space that combines both discrete tokens and continuous concepts, we adopt the RL framework based on Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which stably optimizes the latent policy space and effectively guides the model's internal reasoning process.

Our evaluations on reasoning benchmarks show that SCM improves performance across different model scales. Furthermore, our approach demonstrates high representational stability; Principal Component Analysis (PCA) of hidden states reveals only slight latent shifts, comparable to the GRPO baseline. In summary, our main contributions are as follows:

- We propose SCM, a lightweight framework that integrates soft concepts into hidden states and utilizes reinforcement learning to dynamically optimize these continuous representations.
- Our evaluations on reasoning benchmarks show that SCM consistently improves performance across different model scales and architectures.
- The analysis of computational efficiency and training stability reveals that SCM demonstrates stability during training and incurs marginal inference overhead, offering a highly favorable trade-off for practical deployment.

2 Related work

2.1 Latent reasoning

Training-based Methods. Coconut (Hao et al., 2024) leverages the model's hidden states as continuous latent vectors for reasoning. By directly utilizing the hidden state from the previous step as the input embedding for the subsequent step, it achieves more flexible and efficient reasoning within the latent space. However, this approach requires a fixed Chain-of-Thought (CoT). Despite its potential, Coconut relies heavily on Ground Truth CoT data and necessitates a complex multi-stage training strategy to internalize the process from explicit to implicit reasoning. This cumbersome training procedure significantly increases computational overhead, thereby limiting its scalability in large-scale scenarios. CoCoMix (Tack et al., 2025), on the other hand, employs Sparse Autoencoders (SAE) to extract "continuous concepts." By interleaving these concepts with the model's hidden states, it effectively enhances the model's semantic representation capabilities.

Inference-time Methods. Soft Thinking (Zhang et al., 2025) constructs "concept tokens" via the probability-weighted averaging of token embeddings, enabling the model to explore multiple reasoning paths simultaneously. However, Wu et al. (Wu et al., 2025) observed that despite the inclusion of soft tokens, the distribution tends to rapidly collapse towards the top-1 token, thereby diminishing the benefit of soft tokens in conveying richer information. To address this limitation, the authors proposed Stochastic Soft Thinking, a method that restores diversity by incorporating randomness via Dirichlet or Gumbel-Softmax sampling. Similarly, Mixture of Inputs (MOI) (Zhuang et al., 2025) synthesizes the probability distribution predicted in the previous step with the actually sampled token using Bayesian estimation to generate a novel input representation. This design allows the model to preserve richer internal representations throughout the entire reasoning process.

2.2 Reinforcement Learning

Reinforcement Learning (RL) has become central to optimizing reasoning capabilities, shifting focus from imitating human labels (SFT) to optimizing outcomes. Early approaches like Proximal Policy Optimization (PPO) (Schulman et al., 2017) employ actor-critic architectures that jointly train a state value function V for advantage estimation, a design that inevitably incurs high memory overheads.

To address this, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) emerged as a more efficient, critic-less alternative. By estimating base-

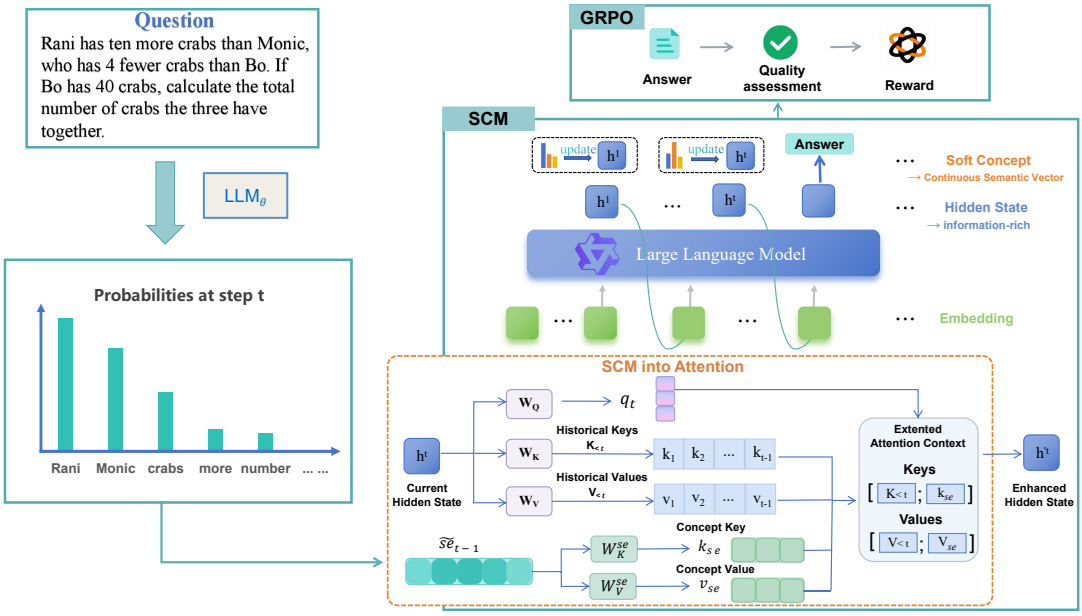

Figure 1: Training Framework for Soft Concept Mixing (SCM).

lines directly from group scores rather than relying on a separate value network, GRPO significantly reduces computational overhead. It stabilizes training by directly optimizing the relative advantages of reasoning trajectories within a sampled group, effectively reinforcing reasoning capability.

Hybrid Reasoning Policy Optimization (HRPO) (Yue et al., 2025) introduces a reinforcement learning framework designed to enhance latent reasoning. Unlike traditional methods that rely solely on discrete tokens, HRPO integrates continuous hidden states into the generation process via a learnable gating mechanism. This mechanism gradually incorporates information from prior hidden states into the current token embedding, allowing the model to leverage a "hybrid" representation. However, this architectural addition incurs extra computational overhead due to the gating parameters. Furthermore, achieving optimal performance is non-trivial, as it necessitates extensive experimentation to precisely tune the hyperparameters that govern the balance between discrete and continuous representations during RL training.

These standard RL approaches excel at optimizing reasoning strategies based on outcomes. However, they are often constrained by the rigid discreteness of language generation, which forces the model to make hard token commitments at every step, potentially discarding valuable alternative reasoning paths that could enhance overall solution quality.

3 Methodology

In this section, we introduce Soft Concept Mixing (SCM), a lightweight training framework designed to enhance the latent reasoning capabilities of Large Language Models. The overall process is illustrated in Figure 1.

3.1 Soft Concept Vector Generation

To generate a guidance signal more informative than discrete tokens, we introduce the soft concept vector. At step t, given the output logits $\mathbf{z}_t \in \mathbb{R}^{|V|}$, we compute the probability distribution $p_t = \text{Softmax}(\mathbf{z}_t)$. The soft concept vector $\hat{\boldsymbol{se}}_t$ is then constructed as the expectation of token embeddings E over this distribution:

$$\hat{\boldsymbol{se}}_t = \sum_{i=1}^{|V|} p_{t,i} \cdot e(x_i) = \mathbb{E}_{x \sim p_t}[e(x)].$$

In practice, we apply top-k and top-p truncation to suppress low-probability noise and concentrate probability mass on semantically relevant candidates, followed by renormalization. To ensure semantic consistency, these truncation settings are aligned with the decoding configurations used during inference. The resulting vector $\hat{\boldsymbol{se}}_t$ provides a continuous summary of the high-probability token embeddings, capturing the dominant distributional concept at the current step.

3.2 Soft Concept Mixing

To effectively inject the rich distributional information carried by the soft concept vector into the model's latent representation space, we leverage the native self-attention mechanism of the Transformer. We project the soft concept into the Key-Value subspace across all transformer layers and all attention heads, providing it as additional context.

Let $\boldsymbol{h}_t^{(l)}$ denote the hidden state input of the l-th layer at time step t. In standard self-attention computation, the model first derives the Query vector, along with the Key and Value matrices representing the historical context:

$$\boldsymbol{q}_t^{(l)} = \boldsymbol{h}_t^{(l)} \boldsymbol{W}_Q^{(l)}, \quad \boldsymbol{K}_{<t}^{(l)}, \quad \boldsymbol{V}_{<t}^{(l)}$$

where $\boldsymbol{W}_Q^{(l)}$ is the projection matrix, and $\boldsymbol{K}_{<t}^{(l)}$ and $\boldsymbol{V}_{<t}^{(l)}$ denote the sequences of Keys and Values from previous time steps.

To inject the soft concept vector $\hat{\boldsymbol{se}}_{t-1}$ generated from the previous time step, we first project it into the current layer's semantic space to generate the **concept Key** and **concept Value**:

$$\boldsymbol{k}_{\text{se}}^{(l)} = \hat{\boldsymbol{se}}_{t-1} \boldsymbol{W}_K^{(l)}, \quad \boldsymbol{v}_{\text{se}}^{(l)} = \hat{\boldsymbol{se}}_{t-1} \boldsymbol{W}_V^{(l)}$$

Subsequently, we extend the attention context with these concept vectors. The modified Key and Value sequences become:

$$\hat{\boldsymbol{K}}^{(l)} = [\boldsymbol{K}_{<t}^{(l)}; \boldsymbol{k}_{\text{se}}^{(l)}], \quad \hat{\boldsymbol{V}}^{(l)} = [\boldsymbol{V}_{<t}^{(l)}; \boldsymbol{v}_{\text{se}}^{(l)}]$$

Finally, the model computes the attention output based on this extended context:

$$\boldsymbol{h}_t'^{(l)} = \text{Attention}(\boldsymbol{q}_t^{(l)}, \hat{\boldsymbol{K}}^{(l)}, \hat{\boldsymbol{V}}^{(l)})$$

Thus, the soft concept $\hat{\boldsymbol{se}}_{t-1}$ participates in the reasoning process as a contextual guidance signal. Crucially, this approach ensures geometric consistency between the concept vectors and the model's original feature space, while empowering the model to flexibly "backtrack" to utilize distribution information within the latent space.

3.3 Optimizing Latent Reasoning via RL

To effectively optimize the reasoning process, we define the generation process as a discrete token policy augmented by a latent continuous concept policy. The discrete policy governs token selection, while the latent continuous policy generates soft concept vectors that enhance the model's hidden state, bridging continuous semantic exploration and discrete decision-making.

We formalize the reasoning process as a Markov Decision Process (MDP). At each step t, the state s_t consists of the input $\boldsymbol{x}$, the history of generated tokens $\boldsymbol{y}_{<t}$, the current hidden state $\boldsymbol{h}_t$, and the previous concept vector $\hat{\boldsymbol{se}}_{t-1}$. The policy π_θ operates in two sequential phases: 1) **Discrete Action:** The model integrates $\hat{\boldsymbol{se}}_{t-1}$ into the attention context to derive the enhanced hidden state $\boldsymbol{h}_t'$, and subsequently samples the discrete token y_t from the distribution $\boldsymbol{p}_t = \pi_\theta(\cdot \mid \boldsymbol{h}_t')$. 2) **Latent Action:** Using the materialized distribution $\boldsymbol{p}_t$, a deterministic operation computes the current soft concept vector $\hat{\boldsymbol{se}}_t = \mathbb{E}_{v \sim \boldsymbol{p}_t}[\boldsymbol{e}(v)]$ via a differentiable weighted sum over the vocabulary.

GRPO Training. We optimize π_θ using GRPO (Shao et al., 2024). For each input $\boldsymbol{x}$, we sample K rollouts. In each rollout, the model applies the concept-mixing operation at every step to obtain the enhanced state $\boldsymbol{h}_t'$ and then samples $y_t \sim \pi_\theta(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{h}_t')$. We compute a total reward $r^{(k)}$ per rollout based on correctness and formatting constraints. Using the group-normalized advantage $A^{(k)} = (r^{(k)} - \mu)/(\sigma + \epsilon)$, our training objective is to minimize:

$$\mathcal{L}(\theta) = -\mathbb{E}_x \Bigg[\frac{1}{K} \sum_{k=1}^{K} \frac{1}{|y^{(k)}|} \sum_{t=1}^{|y^{(k)}|} \Bigg(\min \Big(\rho_t^{(k)} A^{(k)}, \text{clip} (\rho_t^{(k)}, 1-c, 1+c) A^{(k)} \Big) - \beta D_{\text{KL}}(\pi_\theta \parallel \pi_{\text{ref}}) \Bigg) \Bigg]$$

where $\rho_t^{(k)} = \frac{\pi_\theta(y_t^{(k)} | \boldsymbol{x}, \boldsymbol{y}_{<t}^{(k)}, \boldsymbol{h}_t'^{(k)})}{\pi_{\text{old}}(y_t^{(k)} | \boldsymbol{x}, \boldsymbol{y}_{<t}^{(k)}, \boldsymbol{h}_t'^{(k)})}$ denotes the probability ratio, c is the clipping threshold, and π_{ref} is the reference model. This eliminates the need for a separate value function, providing stable gradients to directly reinforce high-reward trajectories.

Continuous Semantic Conditioning. Instead of treating the discarded probability distribution p_t as a byproduct, SCM leverages it to enrich the immediate downstream state. By deterministically injecting the expected semantic vector $\hat{se}_t$ into the KV cache, the subsequent state h'_{t+1} is conditioned on a multi-hypothesis representation rather than a single, hard-committed discrete token. Consequently, when GRPO optimizes the discrete token policy based on outcome rewards, it operates over these semantically enriched states, allowing the model to make more robust decisions grounded in broader latent context.

Model	Method	MATH500	AIME24	GSM8K	GPQA-D	MMLU	BBH	Avg.
DS-R1-Q-7B	CoT	91.75	50.12	90.25	50.45	66.58	45.15	65.72
	Soft Thinking	90.55	48.93	89.40	49.62	66.35	46.20	65.18
	GRPO	93.20	53.13	90.45	51.01	67.03	47.30	67.02
	SCM	**94.40**	**56.17**	**92.53**	**52.85**	**67.91**	**48.90**	**68.79**
	w/o hidden states	93.00	54.58	90.80	51.10	66.90	45.25	66.94
DS-R1-L-8B	CoT	81.09	44.60	69.35	44.40	71.15	46.30	59.48
	Soft Thinking	77.56	40.50	70.15	42.85	70.60	47.90	58.26
	GRPO	81.42	47.85	70.25	44.90	72.10	48.05	60.76
	SCM	**82.60**	**54.85**	**71.90**	**45.58**	**73.90**	**48.90**	**62.96**
	w/o hidden states	81.90	50.50	70.60	45.10	71.90	47.10	61.18
DS-R1-Q-1.5B	CoT	80.80	30.00	75.20	33.30	44.38	24.90	48.10
	Soft Thinking	80.60	20.00	77.30	31.82	44.17	26.20	46.68
	GRPO	82.75	33.33	76.95	33.73	45.02	25.60	49.56
	SCM	**84.00**	**36.52**	**77.79**	**35.98**	**46.90**	**29.40**	**51.77**
	w/o hidden states	80.60	33.33	77.18	33.84	45.80	26.10	49.48
Qwen2.5-7B-INS	CoT	75.00	12.08	89.61	32.58	68.23	59.10	56.10
	Soft Thinking	76.10	13.33	90.96	32.20	68.27	60.00	56.81
	GRPO	76.20	13.33	90.42	32.96	68.86	60.65	57.07
	SCM	**77.60**	**18.67**	**91.89**	**33.43**	**70.11**	**63.40**	**59.18**
	w/o hidden states	75.80	13.33	90.37	32.32	68.09	61.00	56.82

Table 1: Accuracy of SCM compared to other baselines on six reasoning benchmarks. The GRPO baseline enhances CoT with RL fine-tuning. Best results are in **bold**.

4 EXPERIMENTS

4.1 Experimental setup

Datasets. We train on two mathematical reasoning datasets—GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021), covering a wide difficulty range from elementary arithmetic to competition-level problems. Furthermore, we evaluate our proposed method and all baselines on six public reasoning benchmarks: GSM8K, MATH500, AIME 2024 (MAA Committees, 2024), GPQA-Diamond (Rein et al., 2024), BIG-Bench Hard (BBH) (Suzgun et al., 2023), and MMLU (Hendrycks et al., 2020).

Models. We conduct experiments using four open-source LLMs: DeepSeek-R1-Distill-Qwen-7B, DeepSeek-R1-Distill-Llama-8B, DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025), and Qwen2.5-7B-Instruct (Team et al., 2024).

RL Setup. We fine-tune our models using the GRPO algorithm, implemented with the Unsloth framework. We apply a trainable LoRA module (Hu et al., 2022) (rank=32, alpha=64), targeting all linear layers in the attention (Q/K/V/O) and MLP (gate/up/down) blocks. The models are trained for one epoch on four NVIDIA A100 GPUs with a global batch size of 64, sampling $K = 8$ rollouts per prompt. We use a learning rate of 5×10^{-6} and a maximum output length of 4096 tokens. Comprehensive details regarding the reward formulation and system prompts are provided in Appendix B.

Evaluation Settings. All methods use the same chat templates and CoT prompting, and we report avg@8 accuracy, defined as the average accuracy over 8 independent sampling runs per problem. To assess varying reasoning budgets, extended pass@k and majority-vote@k metrics are detailed in Appendix E. For decoding, we use top-k=30 sampling with temperature 0.6, top-p=0.95, and a maximum generated length of 32,768 tokens. Soft Thinking (Zhang et al., 2025) is evaluated with its official implementation and default hyperparameters, detailed in Appendix C.1.

4.2 Main Results

We evaluated the proposed Soft Concept Mixing (SCM) method on four models, and compared it with traditional Chain-of-Thought (CoT) reasoning, Soft Thinking, and the reinforcement learning baseline of GRPO fine-tuned on CoT. In addition, we conducted an ablation study (denoted as SCM w/o hidden states) where the soft concept vector is directly added to the token embedding of the next step. Detailed results are shown in Table 1. To verify the robustness of these performance improvements, the results of statistical significance

Method	MATH500	AIME 2024	GSM8K	GPQA-Diamond	MMLU	BBH	Avg
CoT Thinking	91.75	50.12	90.25	50.45	66.58	45.15	65.72
Soft Thinking(Gumbel)	93.45	52.08	89.52	50.69	67.35	45.60	66.45
MOI	93.53	52.08	89.04	48.99	66.85	45.95	66.07
HRPO	93.72	55.00	89.88	50.75	66.70	46.30	67.06
Coconut	92.60	43.50	89.25	47.10	66.50	45.20	64.03
SCM (Ours)	**94.40**	**56.17**	**92.53**	**52.85**	**67.91**	**48.90**	**68.79**

Table 2: Comparison with concurrent methods on DeepSeek-R1-Distill-Qwen-7B. We report the avg@8 accuracy (%) across six benchmarks.

tests are provided in Appendix D.

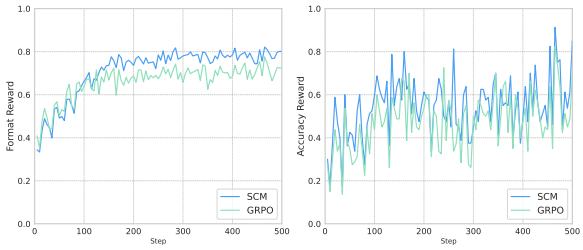

Figure 2: Training process of DeepSeek-R1-Distill-Qwen-1.5B on GRPO utilizing our proposed SCM method.

First and foremost, SCM demonstrates consistent efficacy across diverse model architectures. On both DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-Llama-8B, SCM achieves superior average accuracies of 68.79% and 62.96%, respectively, outperforming the CoT and Soft Thinking baselines by significant margins ranging from 3.1% to 4.6%. Furthermore, SCM exhibits superior performance compared to the strong GRPO reinforcement learning baseline, particularly on challenging benchmarks such as AIME 2024 and BBH. This indicates that by integrating soft concepts directly into the reinforcement learning process, SCM enables the model to actively adapt its internal representations to leverage these continuous signals, thereby achieving more robust reasoning trajectories.

Furthermore, the robustness of SCM is validated on models with limited capacity. On the lightweight DeepSeek-R1-Distill-Qwen-1.5B, SCM still surpasses CoT, Soft Thinking, and GRPO. This stability is also reflected in the training dynamics, where Figure 2 shows SCM exhibits higher and more stable rewards compared to GRPO in later training stages. The method's applicability extends beyond reasoning-oriented architectures, as SCM also yields consistent gains on an instruction-tuned model Qwen2.5-7B-Instruct. Finally, our ablation study confirms the necessity of the hidden state fusion component, as its removal degrades performance.

4.3 Comparison with Concurrent Works

We evaluate SCM against recent training-based (Coconut (Hao et al., 2024), HRPO (Yue et al., 2025)) and inference-time (Stochastic Soft Thinking (Wu et al., 2025), MOI (Zhuang et al., 2025)) latent approaches. For fairness, all baselines adopt SCM's exact configurations (Appendix C). Table 2 reports the main results. Extended evaluations (Appendix F) show SCM consistently outperforms concurrent works CODI (Shen et al., 2025) and LTPO (Ye et al., 2025).

Compared to existing latent approaches, SCM is a purpose-built mechanism for autonomous latent exploration. Unlike Coconut, which relies on rigid imitative supervision and a computationally expensive multi-stage curriculum, SCM leverages outcome-driven RL to autonomously discover intuition signals without predefined trajectories. Furthermore, SCM circumvents the architectural complexity of HRPO by projecting concepts natively into the Key-Value subspace, preserving geometric consistency without specialized gating modules. Unlike inference-only strategies (e.g., Stochastic Soft Thinking, MOI) that inherently suffer from training-inference mismatch, SCM integrates soft concepts directly during training. Crucially, SCM establishes a continuous semantic pathway, enabling later steps to be dynamically conditioned on the rich distributions generated in earlier steps.

Collectively, SCM maintains robust out-of-domain performance without catastrophic forgetting, exhibiting distinct advantages in reasoning capability, architectural simplicity, and autonomous optimization over existing latent paradigms.

Figure 3: PCA shift of DeepSeek-R1-Distill-Llama-8B across six datasets

5 Analysis

5.1 Latent Space Shift Analysis

To further investigate the impact of SCM on the internal representation structure of models, we employ Principal Component Analysis (PCA) to visualize and quantitatively evaluate the shift of latent representation centers before and after training across different models (Huan et al., 2025; Zheng et al., 2025; Xu et al., 2025a). Specifically, we apply PCA ($n = 2$) to the hidden states of all transformer layers. For each layer l and evaluation dataset $\mathcal{D}$, we compute the centroid of the hidden states $z_{l,\mathcal{D}}$. The latent space deviation is then defined as the Euclidean distance between the centers of the trained and initialized models: $d_{l,\mathcal{D}} = \|z_{l,\mathcal{D}}^{(\text{trained})} - z_{l,\mathcal{D}}^{(\text{init})}\|_2$. The quantitative results in Table 3 are derived by averaging $d_{l,\mathcal{D}}$ across all transformer layers and all datasets within each specific category, providing a robust measure of global representational drift.

Model	Method	Math Datasets	Knowledge Datasets
DS-R1-L-8B	GRPO	0.02	0.07
	SCM	0.09	0.05
DS-R1-Q-1.5B	GRPO	0.04	0.06
	SCM	0.05	0.08

Table 3: PCA shift distances of representation centers across models.

Figure 3 presents PCA distribution comparisons for the same model evaluated across six datasets. We observe that whether trained solely with GRPO or with the additional SCM mechanism, the overall latent representation shows no significant large-scale drift, indicating stable policy optimization. This stability is quantitatively confirmed across different model architectures in Table 3. Crucially, SCM achieves significant performance gains despite this minimal representational shift. This suggests that the soft concepts facilitate localized adjustments to intermediate representations, rather than inducing large-scale shifts in the overall representation space.

5.2 Ablation on the Added Noise

To determine whether the performance gains of SCM stem from the preservation of multi-hypothesis semantic distributions, or merely from stochastic regularization and dense architectural connectivity within the embedding manifold, we evaluate four information carriers:

- **Gaussian Noise**: A random vector $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma)$ matching SCM's average magnitude, testing pure stochastic regularization.
- **Random Token Mixture**: An average embedding of uniformly sampled tokens matching the candidate set size, injecting structurally valid but semantically irrelevant noise.
- **Logit Concatenation**: The raw, unnormalized logit vector $\boldsymbol{z}_t \in \mathbb{R}^{|V|}$ concatenated via a linear layer, exploring raw feature transmission.
- **Hard Token Embedding (Sampled)**: The exact embedding of the discretely sampled token y_t, providing a contextually accurate but strictly singular semantic signal to isolate the architectural skip-connection effect.

Method	Information Carrier	Integration	GSM8K	AIME2024
GRPO (Baseline)	None	-	90.45	53.13
Gaussian Noise	Random $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma)$	Add to $\boldsymbol{h}_t$	90.52	53.20
Random Token Mix	Random Embeddings $\in \mathbb{R}^d$	Add to $\boldsymbol{h}_t$	89.95	52.40
Logit Concat	Raw Logits $\in \mathbb{R}^{\lvert V \rvert}$	Concat to Input	89.80	51.50
Hard Token Embedding	Sampled Embed $\in \mathbb{R}^d$	Add to $\boldsymbol{h}_t$	91.20	54.35
SCM (Ours)	**Soft Concept** $\in \mathbb{R}^d$	Add to $\boldsymbol{h}_t$	**92.53**	**56.17**

Table 4: Ablation of information carriers on GSM8K and AIME2024 using DeepSeek-R1-Distill-Qwen-7B. We compare SCM against alternative injection strategies to isolate the impact of multi-hypothesis semantic distributions.

As shown in Table 4, injecting arbitrary continuous signals (Gaussian Noise, Random Token Mixture) or unnormalized dense features (Logit Concatenation) fails to improve upon the vanilla GRPO baseline, and often degrades performance due to semantic conflict or the curse of dimensionality. This confirms that simply adding structural perturbations or raw features is counterproductive; the hidden state requires semantically aligned guidance.

Crucially, while the Hard Token Embedding baseline provides a contextually valid residual signal and slightly stabilizes reasoning, it strictly underperforms SCM. This empirical gap definitively validates our core hypothesis: SCM's superiority is not a byproduct of denser cross-layer connectivity, but stems directly from preserving the alternative semantic trajectories encoded within the continuous probability distribution p_t.

Method	Inference (A100)		Training (1 Epoch)	
	Speed	Overhead	Avg@8	Time
Vanilla GRPO	1,584 tok/s	-	67.02%	∼52 h
SCM (Ours)	1,511 tok/s	+4.6%	**68.79%**	∼54 h

Table 5: Efficiency comparison on DeepSeek-R1-Distill-Qwen-7B.

5.3 Computational Efficiency Analysis

We benchmarked the computational cost using DeepSeek-R1-Distill-Qwen-7B on a single NVIDIA A100. As shown in Table 5, SCM introduces a marginal inference overhead of ∼4.6% (1,511 vs. 1,584 tokens/s). Leveraging the truncation strategy, the resulting computation is implemented as a highly parallelized matrix-vector product, ensuring that the per-step overhead remains minimal on GPUs. Regarding training, SCM requires slightly more wall-clock time (∼54h vs. ∼52h per epoch) but yields a significant performance gain (+1.41% Avg@8). This demonstrates that SCM offers a highly favorable trade-off, exchanging minor computational cost for substantial improvements in latent reasoning.

6 CONCLUSION

In this work, we introduced Soft Concept Mixing (SCM), a lightweight training framework that mixes the rich semantic information from the discarded probability distribution back into the model's latent space, and explicitly preserves these continuous distributional signals via reinforcement learning. Our experiments across multiple benchmarks and architectures demonstrate that this approach consistently improves performance across diverse reasoning benchmarks while maintaining stable training dynamics, offering a computationally efficient strategy for enhancing latent reasoning in LLMs.

Limitations

While SCM demonstrates significant improvements in latent reasoning, several limitations warrant future investigation. First, our method has been validated exclusively within the GRPO framework; it remains to be verified whether other algorithms (e.g., PPO) or standard Supervised Fine-Tuning (SFT) can effectively align soft concept distributions. Second, future research could investigate whether latent reasoning is more concentrated within specific semantic blocks or hierarchical layers, and explore the potential expansion from token-level soft concepts to more comprehensive context-level representations. Finally, although the injected soft concept vectors empirically enhance performance by serving as effective "intuition" signals, these continuous biases are not directly interpretable by humans, and we currently lack qualitative analysis to fully characterize the specific semantic nature of these latent injections.

References

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jingcheng Deng, Liang Pang, Zihao Wei, Shichen Xu, Zenghao Duan, Kun Xu, Yang Song, Huawei Shen, and Xueqi Cheng. 2025. Latent reasoning in llms as a vocabulary-space superposition. *arXiv preprint arXiv:2510.15522*.

Evelina Fedorenko and Rosemary Varley. 2016. Language and thought are not the same thing: evidence from neuroimaging and neurological patients. *Annals of the New York Academy of Sciences*, 1369(1):132–153.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: Program-aided language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10764–10799. PMLR.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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. *arXiv preprint arXiv:2412.06769*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. 2025. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*.

Elita Lobo, Chirag Agarwal, and Himabindu Lakkaraju. 2025. On the impact of fine-tuning on chain-of-thought reasoning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11679–11698.

MAA Committees. 2024. American invitational mathematics examination (aime). https://maa.org/maa-invitational-competitions/. Accessed: 2024-02-10.

MAA Committees. 2025. American invitational mathematics examination (aime). https://maa.org/maa-invitational-competitions/. Accessed: 2025-03-10.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pages 975–984.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2080–2094.

R Quian Quiroga, Leila Reddy, Gabriel Kreiman, Christof Koch, and Itzhak Fried. 2005. Invariant visual representation by single neurons in the human brain. *Nature*, 435(7045):1102–1107.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1743–1752, Lisbon, Portugal. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

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. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 677–693.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and 1 others. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Jihoon Tack, Jack Lanchantin, Jane Yu, Andrew Cohen, Ilia Kulikov, Janice Lan, Shibo Hao, Yuandong Tian, Jason Weston, and Xian Li. 2025. Llm pretraining with continuous concepts. *arXiv preprint arXiv:2502.08524*.

Qwen Team and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3).

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Junhong Wu, Jinliang Lu, Zixuan Ren, Gangqiang Hu, Zhi Wu, Dai Dai, and Hua Wu. 2025. Llms are single-threaded reasoners: Demystifying the working mechanism of soft thinking. *arXiv preprint arXiv:2508.03440*.

Xiaoyu Xu, Xiang Yue, Yang Liu, Qingqing Ye, Huadi Zheng, Peizhao Hu, Minxin Du, and Haibo Hu. 2025a. Unlearning isn't deletion: Investigating reversibility of machine unlearning in llms. *arXiv preprint arXiv:2505.16831*.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. 2025b. SoftCoT: Soft chain-of-thought for efficient reasoning with LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23336–23351, Vienna, Austria. Association for Computational Linguistics.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024. Do large language models latently perform multi-hop reasoning? *arXiv preprint arXiv:2402.16837*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Wengao Ye, Yan Liang, and Lianlei Shan. 2025. Thinking on the fly: Test-time reasoning enhancement via latent thought policy optimization. *arXiv preprint arXiv:2510.04182*.

Zhenrui Yue, Bowen Jin, Huimin Zeng, Honglei Zhuang, Zhen Qin, Jinsung Yoon, Lanyu Shang, Jiawei Han, and Dong Wang. 2025. Hybrid latent reasoning via reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, Shuohang Wang, Yelong Shen, and Xin Eric Wang. 2025. Soft thinking: Unlocking the reasoning potential of llms in continuous concept space. *arXiv preprint arXiv:2505.15778*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. 2025. Spurious forgetting in continual learning of language models. *arXiv preprint arXiv:2501.13453*.

Yufan Zhuang, Liyuan Liu, Chandan Singh, Jingbo Shang, and Jianfeng Gao. 2025. Text generation beyond discrete token sampling. *arXiv preprint arXiv:2505.14827*.

A Benchmark Details

We utilize six benchmarks to evaluate reasoning across diverse domains and difficulty levels:

- **GSM8K** (Cobbe et al., 2021): A dataset of high-quality grade school math problems, containing 7.5k training and 1.3k test samples.
- **MATH500** (Hendrycks et al., 2021): A collection of challenging high school competition problems across seven disciplines (e.g., Algebra, Geometry), consisting of 7,500 training and 500 test samples.
- **AIME 2024** (MAA Committees, 2024): Consists of advanced competition-level problems sourced from the 2024 American Invitational Mathematics Examination.
- **GPQA-Diamond** (Rein et al., 2024): A graduate-level benchmark requiring deep domain knowledge and complex reasoning, primarily in STEM fields.
- **BIG-Bench Hard (BBH)** (Suzgun et al., 2023): A suite of 23 challenging tasks selected from Big-Bench to evaluate complex reasoning capabilities.
- **MMLU** (Hendrycks et al., 2020): A massive multitask benchmark covering 57 subjects across STEM, humanities, and social sciences.

B Experimental Details

In this section, we provide the specific configurations used in our experiments, including the reward formulation and system prompts.

B.1 Reward Function Formulation

To guide the reinforcement learning process, we design a composite reward function $r^{(k)}$ that consists of two components: a correctness reward r_{acc} and a format compliance reward r_{fmt}.

- **Correctness Reward (r_{acc}):** This binary term is set to 1 if the model's final answer matches the ground truth, and 0 otherwise.
- **Format Reward (r_{fmt}):** To ensure the structural integrity of the reasoning trace, we grant $+0.25$ for the presence of each of the four required structural tags: `<think>`, `</think>`, `<answer>`, and `</answer>`.

B.2 RL Training Prompt

The system prompt used during the Reinforcement Learning (RL) training phase is presented below:

System Prompt for RL Training

Think about the problem and provide your working out. Then put your final answer within `\boxed{}`. The reasoning process and answer are enclosed within '`<think>`' '`</think>`' and '`<answer>`' '`</answer>`' tags, respectively, i.e., `<think>` {reasoning process} `</think>` `<answer>` `\boxed{{final answer}}` `</answer>`.
{Question}

B.3 Mathematical Reasoning Prompts

The evaluation prompt for mathematical reasoning benchmarks (GSM8K, MATH, AIME 2024) is as follows:

System Prompt for Math Benchmarks

Please reason step by step, and put your final answer within `\boxed{{}}`.
{Question}

B.4 Knowledge-Based Reasoning Prompts

The evaluation prompt for knowledge-based reasoning benchmarks (MMLU, GPQA, BBH) is as follows:

System Prompt for Knowledge-Based Benchmarks

Please solve the following multiple-choice question. Please show your choice in the answer field with only the choice letter, e.g.,"answer": "C".
{Question}

C Implementation Details of Concurrent Baselines

To ensure a fair and rigorous comparison, we strictly followed the official training protocols and hyperparameter settings for all concurrent baselines. For training-based methods (Coconut and HRPO), models were trained on the standard splits of GSM8K and MATH. For inference-time strategies (Soft Thinking, Stochastic Soft Thinking, and MOI), we kept the evaluation setup identical to our SCM runs (using the same chat template, decoding temperature 0.6, maximum output lengths, and avg@8 metric).

C.1 Soft Thinking

We evaluate Soft Thinking (Zhang et al., 2025) using its official codebase. Following the default configuration, the concept tokens are determined using a sampling strategy with temperature 0.6, top-$p = 0.95$, and top-$k = 30$. Regarding the inference termination strategy, we utilize the "Cold Stop" mechanism with an entropy threshold of 0.1 and a length threshold of 256 tokens.

C.2 Coconut Reimplementation

Since the original Coconut (Hao et al., 2024) implementation targeted GPT-2, we adapted the codebase for the 7B model using Low-Rank Adaptation (LoRA) to manage computational costs while maintaining performance. We strictly implemented its signature **multi-stage continuous reasoning curriculum**:

- **Stage 0 (Warm-up):** The model is first trained with standard Supervised Fine-Tuning (SFT) on the target logic chains.

Model	Method	MATH500	AIME24	GSM8K	GPQA-D	MMLU	BBH	Avg.
DS-R1-Q-7B	GRPO	$93.12_{\pm 0.42}$	$52.85_{\pm 0.61}$	$90.30_{\pm 0.32}$	$51.15_{\pm 0.48}$	$67.10_{\pm 0.22}$	$47.12_{\pm 0.65}$	$66.94_{\pm 0.28}$
	SCM (Ours)	$\mathbf{94.28}^{*}_{\pm 0.28}$	$\mathbf{55.90}^{*}_{\pm 0.52}$	$\mathbf{92.41}^{*}_{\pm 0.28}$	$\mathbf{52.70}^{*}_{\pm 0.45}$	$67.85_{\pm 0.25}$	$48.75_{\pm 0.58}$	$\mathbf{68.65}^{*}_{\pm 0.24}$
DS-R1-L-8B	GRPO	$81.25_{\pm 0.48}$	$47.60_{\pm 0.62}$	$70.10_{\pm 0.45}$	$44.95_{\pm 0.55}$	$72.05_{\pm 0.32}$	$48.15_{\pm 0.68}$	$60.68_{\pm 0.38}$
	SCM (Ours)	$\mathbf{82.45}^{*}_{\pm 0.35}$	$\mathbf{54.60}^{*}_{\pm 0.58}$	$\mathbf{71.75}^{*}_{\pm 0.32}$	$45.65_{\pm 0.42}$	$73.80_{\pm 0.28}$	$48.80_{\pm 0.55}$	$\mathbf{62.84}^{*}_{\pm 0.30}$

Table 6: Significance test of SCM. Values are Mean $\pm$ SD across 3 evaluation seeds. **Bold*** denotes $p < 0.05$ (paired t-test).

- **Curriculum Stages:** Following the initial warm-up, we progressively increase the number of continuous latent tokens (k) that replace the original discrete reasoning tokens.

- **Training Schedule:** Each stage consists of 3 training epochs. The transition between stages allows the model to gradually adapt from discrete CoT generation to continuous latent reasoning, ensuring the stability of the latent space.

C.3 HRPO Settings

For Hybrid Reasoning Policy Optimization (HRPO) (Yue et al., 2025), we follow the official implementation and hyperparameter configurations described in the original work.

Gating Mechanism. HRPO introduces a learnable gating parameter Λ to balance latent hidden representations and token embeddings. Following the original setup, the initialization is chosen such that the hidden ratio falls within the recommended range $[0.95, 0.999]$. Sensitivity analysis in the HRPO study shows that performance remains stable across this interval with only minor variations across knowledge and STEM benchmarks. In our experiments, we adopt $\Lambda = 0.99$, which lies within this effective range and provides competitive performance across tasks.

C.4 Stochastic Soft Thinking

Stochastic Soft Thinking (Wu et al., 2025) utilizes **Gumbel-Softmax sampling** to inject stochastic noise into the token logits prior to normalization, thereby diversifying the mixture of tokens. In our experiments, we set the internal **Gumbel-Softmax distribution hyperparameter** to $\tau = 0.5$, following the optimal configuration reported in the original study to provide a continuous approximation for the categorical distribution.

C.5 Mixture of Inputs (MOI)

Mixture of Inputs (MOI) (Zhuang et al., 2025) blends the discrete sampled token with the discarded token distribution to maintain richer information.The hyperparameter β controls the concentration over the mixing weights, effectively balancing the model's prior belief (the output distribution) with its observed evidence (the sampled token). We set the concentration parameter to $\beta = 1$.

D Statistical Significance Analysis

In our main experiments (Table 1), we follow the standard practice of reporting the `Avg@8` evaluation metric to evaluate the reasoning capabilities of the models. To strictly evaluate the statistical significance of the performance gains achieved by SCM, we conduct additional sampling-based evaluations. Specifically, utilizing the exact same model checkpoints, we evaluate the DS-R1-Q-7B and DS-R1-L-8B backbones across 3 different random seeds during the decoding phase.

Averaged results and standard deviations ($\pm\sigma$) across these 3 independent evaluation runs are presented in Table 6. Furthermore, we perform a paired t-test between the trial outcomes of SCM and the strongest baseline (GRPO). Results demonstrating statistically significant improvements ($p < 0.05$) are highlighted in **bold** with an asterisk (*).

These findings confirm that SCM consistently outperforms standard RL optimization. The analysis demonstrates that SCM delivers reliable and highly significant performance gains on reasoning-intensive benchmarks, exhibiting strong statistical confidence without relying on favorable sampling variance.

E Extended Evaluation: Pass@1 and Majority Voting

To provide a more comprehensive and granular assessment of SCM's performance under varied inference constraints, we report the pass@1 and

Model	Method	Metric	MATH	AIME 24	GSM8k	GPQA-D	MMLU	BBH	Avg.
DS-R1-Q-1.5B	GRPO	pass@1	81.40	33.33	76.95	33.84	45.10	24.50	49.19
	SCM	pass@1	**82.8**	**36.67**	**77.79**	**34.34**	**45.80**	**28.20**	**50.93**
	GRPO	maj@8	84.20	36.67	77.10	34.50	46.20	26.20	50.81
	SCM	maj@8	**85.20**	**36.67**	**78.20**	**36.20**	**47.20**	**30.10**	**52.26**
Qwen2.5-7B-INS	GRPO	pass@1	75.40	13.3	90.67	32.32	68.10	60.56	56.73
	SCM	pass@1	**76.20**	**16.67**	**91.89**	**32.83**	**69.50**	**62.94**	**58.34**
	GRPO	maj@8	78.60	16.67	92.40	34.50	71.20	64.10	59.58
	SCM	maj@8	**80.20**	**20.00**	**94.54**	**35.20**	**72.80**	**65.40**	**61.36**

Table 7: Performance Comparison of Pass@1 and Majority-vote@8.

Method	GSM8K	Math500	ASDiv Aug	AIME24	AIME25	Avg.
LTPO†	88.17	73.60	92.20	**16.67**	**13.33**	56.79
SCM (Ours)	**91.89**	**76.20**	**92.80**	**16.67**	**13.33**	**58.18**

Table 8: Comparison with LTPO on Qwen2.5-7B-Instruct. The symbol † indicates the pass@1 accuracy reported by (Ye et al., 2025). Best results are in bold.

majority-vote@8 (maj@8) results for DeepSeek-R1-Distill-Qwen-1.5B and Qwen2.5-7B-Instruct in Table 7. This evaluation examines the robustness of internalized latent reasoning in both single-trial scenarios and high-budget settings where self-consistency is employed.

F Extended Comparison with Latest Baselines

To further substantiate the effectiveness of our proposed approach, we extend our evaluation to include two recently proposed latent reasoning frameworks: CODI (Shen et al., 2025) and LTPO (Ye et al., 2025). Since these works are highly concurrent, we quote their reported performance directly from the original papers (denoted with †) and evaluate SCM under identical evaluation protocols (e.g., adopting the same base models, benchmark datasets, and pass@1 metric) to ensure a fair comparison.

Comparison with LTPO. We compare SCM against LTPO on the Qwen2.5-7B-Instruct backbone across five mathematical reasoning benchmarks: GSM8K (Cobbe et al., 2021), Math500 (Hendrycks et al., 2021), ASDiv Aug (Miao et al., 2020), AIME 2024 (MAA Committees, 2024), and AIME 2025 (MAA Committees, 2025). As shown in Table 8, SCM consistently outperforms LTPO. Specifically, our method achieves an overall average improvement of 1.39%, with notable gains on GSM8K (91.89% vs. 88.17%) and Math500 (76.20% vs. 73.60%).

Comparison with CODI. CODI attempts to compress the chain-of-thought into a continuous space via a self-distillation paradigm. To compare with this approach, we evaluate SCM against CODI using the LLaMA-3.2-1B-Instruct model. Following CODI's evaluation protocol, we measure performance on SVAMP (Patel et al., 2021), GSM8K-Hard (Gao et al., 2023), and MultiArith (Roy and Roth, 2015). As shown in Table 9, SCM outperforms CODI by a clear margin, demonstrating superior continuous reasoning without requiring the complex multi-stage distillation process.

Method	SVAMP	GSM8K-Hard	MultiArith	Avg.
CODI†	42.90	9.90	92.80	48.53
SCM (Ours)	**56.50**	**11.20**	**94.30**	**54.00**

Table 9: Comparison with CODI on LLaMA-3.2-1B-Instruct. The symbol † indicates the pass@1 accuracy reported by (Shen et al., 2025). Best results are in bold.