INFERENCE-TIME RETHINKING WITH LATENT THOUGHT VECTORS FOR MATH REASONING

Deqian Kong[1,2,†], **Minglu Zhao**[1,†], **Aoyang Qin**[3,†], **Bo Pang**[4], **Chenxin Tao**[3],
David Hartmann[2], **Edouardo Honig**[1], **Dehong Xu**[1], **Amit Kumar**[2], **Matt Sarte**[2],
Chuan Li[2], **Jianwen Xie**[2,‡], **Ying Nian Wu**[1,‡]
[1] Department of Statistics and Data Science, UCLA
[2] Lambda Inc, [3] Tsinghua University, [4] Salesforce Research

ABSTRACT

Standard chain-of-thought reasoning generates a solution in a single forward pass, committing irrevocably to each token and lacking a mechanism to recover from early errors. We introduce *Inference-Time Rethinking*, a generative framework that enables iterative self-correction by decoupling declarative *latent thought vectors* from procedural generation. We factorize reasoning into a continuous latent thought vector (what to reason about) and a decoder that verbalizes the trace conditioned on this vector (how to reason). Beyond serving as a declarative buffer, latent thought vectors compress the reasoning structure into a continuous representation that abstracts away surface-level token variability, making gradient-based optimization over reasoning strategies well-posed. Our prior model maps unstructured noise to a learned manifold of valid reasoning patterns, and at test time we employ a Gibbs-style procedure that alternates between generating a candidate trace and optimizing the latent vector to better explain that trace, effectively navigating the latent manifold to refine the reasoning strategy. Training a 0.2B-parameter model from scratch on GSM8K, our method with 30 rethinking iterations surpasses baselines with 10–15× more parameters, including a 3B counterpart. This result demonstrates that effective mathematical reasoning can emerge from sophisticated inference-time computation rather than solely from massive parameter counts.

1 INTRODUCTION

Human cognition fundamentally distinguishes between **declarative knowledge** and **procedural execution**. The declarative system stores facts, episodes, and explicit plans, while the procedural system encodes the skills and routines required to act upon them (Ullman, 2004). This functional separation is mirrored in the brain's neural architecture: for example, Wernicke's area processes semantic representations (the meaning of concepts), whereas Broca's area orchestrates syntactic production (the procedural act of speech) (Friederici, 2011). This separation allows the brain to hold a "thought" in a declarative buffer, inspect it, and revise it before committing to a procedural output. Ideally, all reasoning systems should preserve this distinction: maintaining a separation between the *content* of a thought and the *process* of expressing it.

Modern language models collapse this distinction. In a standard Transformer (Vaswani et al., 2017), the semantic content (declarative) and the syntactic realization (procedural) are entangled within the same parameter matrix, retrieved through a single, irrevocable forward pass. There is no independent locus for the reasoning plan—no latent workspace that can be critiqued or refined. Consequently, errors in early generation steps propagate unchecked, as the model lacks the mechanism to backtrack and rethink.

We propose to restore this cognitive separation via **latent thought vectors**. We factorize reasoning into two components: a prior encoder that maps random noise tokens into structured latent thought vectors $\mathbf{z}$ (declarative: what to reason about), and a decoder that generates token sequences condi-

[†]Equal Contribution, [‡]Equal Advising. Work done while D. K. was an intern at Lambda, Inc.

tioned on $\mathbf{z}$ (procedural: how to verbalize reasoning). Beyond serving as a declarative buffer, latent thought vectors play a role in the *formalization* of language model reasoning: they compress the reasoning structure into a continuous representation that abstracts away surface-level token variability, making gradient-based optimization over reasoning strategies well-posed.

Crucially, our Transformer encoder maps unstructured noise to a learned manifold whose geometry reflects valid reasoning patterns, allowing us to formulate **reasoning as gradient-based optimization in the space of valid thoughts**. We introduce **Inference-Time Rethinking**, a Gibbs-style procedure that alternates between generation and reflection: (1) *Generate:* The decoder produces a candidate reasoning trace conditioned on the current latent thought $\mathbf{z}$. (2) *Reflect:* The input noise is optimized via gradient descent to maximize the likelihood of the generated trace, moving the thought vector toward a region of the manifold that better explains the desired logic. This loop transforms inference from a single-pass commitment into a dynamic optimization trajectory, allowing the model to recover from errors by navigating the latent space.

This framework also embodies the Complementary Learning Systems theory (McClelland et al., 1995). Our dual-rate training scheme mirrors the interplay between fast and slow learning: we perform **fast** optimization of the latent $\mathbf{z}$ for each instance (adapting to the specific problem) and **slow** updates to the model parameters θ (accumulating general schema). At test time, only the fast pathway remains active, enabling the model to invest compute in per-instance refinement.

By scaling this inference-time computation, we achieve significant performance gains without increasing model size. On GSM8K, our 0.2B-parameter model with 30 rethinking iterations achieves 31.5% accuracy, outperforming 10–15× larger models including CoT-SFT on a 3B counterpart (22.7%) and MARCoS-2B (24.1%). The gains extend to out-of-domain benchmarks, reaching 51.5% on SVAMP and 68.0% on MultiArith. These results demonstrate that effective mathematical reasoning can emerge from the ability to iteratively optimize declarative thoughts, rather than solely from larger parameter counts.

Our contributions are:

- A generative framework that decouples declarative content ($\mathbf{z}$) from procedural generation θ. This separation yields efficient learning: a 0.2B model outperforms much larger monolithic architectures even in a single pass, with strong out-of-domain robustness.
- *Inference-Time Rethinking*, a Gibbs-style procedure that iteratively refines the latent thought by alternating between trace generation and latent optimization, enabling the model to self-correct at test time.
- Empirical validation showing that a 0.2B model with rethinking outperforms 10–15× larger baselines, establishing inference-time computation as a highly effective scaling axis orthogonal to parameter count.

2 Method

2.1 Generative Model with Latent Thought Vectors

Let $\mathbf{x} = (x^{(1)}, \ldots, x^{(N)})$ denote a sequence of ground tokens, which may include a question $\mathbf{x}_q$ and reasoning trace $\mathbf{x}_r$. We introduce latent thought vectors $\mathbf{z} \in \mathbb{R}^d$ that guide generation via the joint model

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\alpha(\mathbf{z})\, p_\beta(\mathbf{x}|\mathbf{z}), \tag{1}$$

where $p_\alpha(\mathbf{z})$ is a learnable prior with parameters α, and $p_\beta(\mathbf{x}|\mathbf{z})$ is an autoregressive decoder with parameters β. Here $\theta = (\alpha, \beta)$ denotes all model parameters. Unlike standard language models that condition only on preceding tokens, $\mathbf{z}$ cross-attends to every decoder layer, providing a global conditioning signal throughout generation (Fig. 1).

Prior Model. We parameterize the prior via a transport map:

$$\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \mathbf{z} = U_\alpha(\mathbf{z}_0), \tag{2}$$

where U_α is a Transformer encoder with bidirectional attention (Vaswani et al., 2017). The encoder transforms noise tokens $\mathbf{z}_0$ into structured thought vectors $\mathbf{z}$; posterior inference over $\mathbf{z}$ reduces to

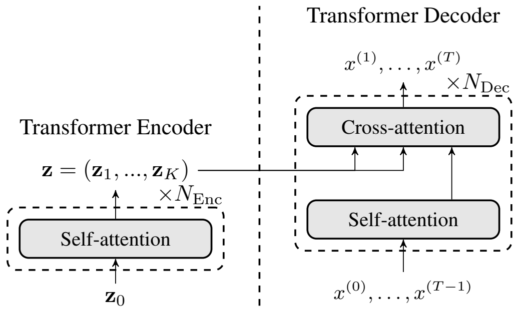

Figure 1: **Model architecture.** The encoder transforms noise tokens $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ into latent thought vectors $\mathbf{z} = U_\alpha(\mathbf{z}_0)$. The decoder generates ground tokens $\mathbf{x}$ conditioned on $\mathbf{z}$ via cross-attention at each layer. Posterior inference operates on $\mathbf{z}_0$ in the base space.

inference over $\mathbf{z}_0$ in the base space. Intuitively, $\mathbf{z}$ represents thought patterns instantiated as multiple tokens, while $\mathbf{z}_0$ indexes into the space of reasoning strategies encoded in the encoder parameters α.

Thought-Guided Generator. The generator $p_\beta(\mathbf{x}|\mathbf{z})$ is a Transformer decoder that incorporates thought vectors at each generation step via cross-attention:

$$p_\beta(\mathbf{x}|\mathbf{z}) = \prod_{n=1}^{N} p_\beta(x^{(n)}|\mathbf{z}, x^{(n-w:n-1)}), \tag{3}$$

where $x^{(n-w:n-1)}$ denotes the w preceding tokens. We use a short context window ($w = 64$) so that the decoder cannot attend directly beyond 64 positions; long-range dependencies must route through $\mathbf{z}$. This architectural bottleneck encourages $\mathbf{z}$ to encode global, sequence-level structure rather than redundant local context.

Our architecture instantiates the encoder-decoder Transformer (Vaswani et al., 2017), repurposed for latent variable modeling. The thought vectors $\mathbf{z}$ serve as instance-specific local parameters inferred per sequence, while $\theta = (\alpha, \beta)$ comprises global parameters learned from all data.

2.2 Learning and Inference

We maximize a variational lower bound on the marginal log-likelihood $\log p_\theta(\mathbf{x}) = \log \int p_\beta(\mathbf{x}|\mathbf{z} = U_\alpha(\mathbf{z}_0))p(\mathbf{z}_0)\, d\mathbf{z}_0$. For each sequence $\mathbf{x}$, we introduce a Gaussian variational posterior over the noise tokens $q(\mathbf{z}_0) = \mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2))$, with local parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ specific to that sequence. The evidence lower bound (ELBO) is:

$$\mathcal{L}(\theta, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) = \mathbb{E}_{q(\mathbf{z}_0)}\left[\log p_\beta(\mathbf{x}|\mathbf{z} = U_\alpha(\mathbf{z}_0))\right] - D_{\text{KL}}\big(q(\mathbf{z}_0)\,\|\,p(\mathbf{z}_0)\big), \tag{4}$$

optimized via the reparameterization trick (Kingma & Welling, 2013), and $p(\mathbf{z}_0) = \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Crucially, $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ is *not* amortized: we do not train an inference network. Instead, each sequence maintains its own $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$, initialized from the prior and optimized directly. This classical variational Bayes approach (Jordan et al., 1999; Blei et al., 2017) avoids the posterior collapse often observed in VAEs (Bowman et al., 2016) for language modeling and provides per-instance precision.

Dual-Rate Optimization. Training alternates between two timescales:

- **Fast local updates**: Optimize $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ for T_{fast} steps using Adam (Kingma & Ba, 2014) with a high learning rate (e.g., 0.3).
- **Slow global updates**: Update $\theta = (\alpha, \beta)$ with a standard learning rate (e.g., 4×10^{-4}).

This dual-rate scheme enables rapid per-instance adaptation of the latent representation while gradually accumulating shared knowledge in the encoder and decoder parameters.

At test time, inference proceeds identically: given a new sequence, we initialize $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ from the prior and run T_{fast} optimization steps. We use $T_{\text{fast}} = 16$ throughout, which we find sufficient for convergence given the model's architectural constraints (short context window, moderate latent dimension). This finite-step inference-time computation is the foundation for the rethinking procedure described next.

Comparison with Latent Thought Models (LTMs). This framework extends Kong et al. (2025), who used a fixed prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and learned only decoder parameters. Our transport-based prior $\mathbf{z} = U_\alpha(\mathbf{z}_0)$ introduces learnable encoder parameters α, yielding a latent space whose geometry is shaped by training. This learned structure proves critical for inference-time rethinking (Section 2.3): gradient-based refinement is more effective when operating in a manifold that reflects the distribution of valid reasoning patterns.

2.3 Inference-Time Rethinking

A standard autoregressive LLM generates a reasoning trace in a single forward pass: $\mathbf{x}_r \sim p_\beta(\mathbf{x}_r|\mathbf{x}_q)$. As inaccurate tokens are committed, errors propagate and the model cannot revisit earlier reasoning steps. We propose *inference-time rethinking*, an iterative procedure that enables revision by alternating between generation and latent reflection.

Given a question $\mathbf{x}_q$, the goal is to sample from the marginal

$$p_\theta(\mathbf{x}_r|\mathbf{x}_q) = \int p_\beta(\mathbf{x}_r|U_\alpha(\mathbf{z}_0), \mathbf{x}_q)\, p(\mathbf{z}_0|\mathbf{x}_q)\, d\mathbf{z}_0 \approx \mathbb{E}_{q(\mathbf{z}_0)}[p_\beta(\mathbf{x}_r|U_\alpha(\mathbf{z}_0), \mathbf{x}_q]. \tag{5}$$

Since direct marginalization is intractable, we employ a Gibbs-style procedure that alternates between trace generation and latent refinement:

$$\text{Step (1) } \textit{Generate:} \quad \mathbf{x}_r^{(t)} \sim p_\beta(\mathbf{x}_r|\mathbf{z}^{(t-1)} = U_\alpha(\mathbf{z}_0^{(t-1)}), \mathbf{x}_q), \tag{6}$$

$$\text{Step (2) } \textit{Reflect:} \quad \mathbf{z}_0^{(t)} = \arg\min_{\mathbf{z}_0} D_{\mathrm{KL}}\big(q(\mathbf{z}_0|\mathbf{x}_r^{(t)}, \mathbf{x}_q) \,\|\, p_\theta(\mathbf{z}_0|\mathbf{x}_r^{(t)}, \mathbf{x}_q)\big). \tag{7}$$

Step (1) generates a candidate reasoning trace given the current latent thought. Step (2) updates the latent to better explain the generated trace, effectively critiquing the thought vectors in latent space. Unlike autoregressive generation, which commits irrevocably to each token, this loop allows the model to **revise past reasoning**: errors in $\mathbf{x}_r^{(t)}$ inform a refined $\mathbf{z}_0^{(t)}$, which then guides a corrected trace $\mathbf{x}_r^{(t+1)}$. In practice, Step (2) is optimized by maximizing the ELBO in Eq. (4) with $(\mathbf{x}_q, \mathbf{x}_r^{(t)})$ as the observed sequence.

We initialize by running variational inference on the question alone to obtain $\mathbf{z}_0^{(0)}$, then iterate Steps (1)–(2) for T_{rethink} rounds. Since the rethinking trajectory explores multiple regions of the latent manifold, we retain the trace with the highest likelihood, and extract the final answer from it.

3 Experiments

3.1 Setup

Training Data. Following Tan et al. (2025), we train all models from scratch on GSM8K-Aug (Deng et al., 2023), an augmented version of GSM8K (Cobbe et al., 2021) containing 385K training examples. Each example pairs a question with an equation-based chain-of-thought solution. Crucially, no model uses pretrained weights, ensuring a fair comparison of learning efficiency rather than transfer from pretraining.

Evaluation Benchmarks. Following Liu et al. (2025), we evaluate on three benchmarks: (1) **GSM8K** (Cobbe et al., 2021), the original test set of 1,319 grade-school math problems (in-domain); (2) **SVAMP** (Patel et al., 2021), 4,138 arithmetic problems constructed via subtle variations in wording and semantic perturbations, testing robustness to surface-form changes (out-of-domain); (3) **MultiArith** (Roy & Roth, 2015), 600 problems from MAWPS (Koncel-Kedziorski et al., 2016) requiring multi-step arithmetic reasoning (out-of-domain). We report accuracy as the evaluation metric.

Baselines. We compare against CoT supervised fine-tuning (CoT-SFT) (Wei et al., 2022) and recent latent reasoning methods: iCoT-SI (Deng et al., 2024), Coconut (Hao et al., 2024), CoLaR (Tan et al., 2025), CODI (Shen et al., 2025), and MARCoS (Liu et al., 2025). Baselines use Qwen2.5-0.5B or Qwen2.5-3B (Team et al., 2024) as backbones; MARCoS uses a 2B model. All baseline numbers are from Liu et al. (2025).

Our Model. We use a 0.2B-parameter Transformer following the Llama-2 (Touvron et al., 2023) architecture with a context window of $w = 64$, a hidden size of 1024, an 8-layer decoder, and a 2-layer encoder with 64 latent tokens. We run $T_{\text{fast}} = 16$ inference steps for latent optimization. Note that our model is smaller than every baseline. We report results for single-pass generation (Rethink-1) and iterative rethinking with $T_{\text{rethink}} = 30$ rounds (Rethink-30) in Table 1.

3.2 Main Results

Table 1: Accuracy (%) on mathematical reasoning benchmarks. All models trained from scratch on equation-format GSM8K-Aug training set. Best in **bold**, second-best underlined.

Backbone	Method	GSM8K	SVAMP	MultiArith
Qwen2.5-0.5B	CoT-SFT	20.31	27.92	41.48
	iCoT-SI	14.81	18.46	21.33
	Coconut	15.27	17.76	15.50
	CoLaR	15.45	23.09	38.60
	CODI	2.55	1.66	1.60
	MARCoS (2B)	24.11	27.77	42.33
Qwen2.5-3B	CoT-SFT	22.70	27.25	50.20
	iCoT-SI	14.08	16.26	19.67
	Coconut	9.95	13.53	8.78
	CoLaR	22.60	25.83	41.83
	CODI	1.24	1.12	1.20
Ours (0.2B)	Rethink-1	<u>25.93</u>	<u>47.37</u>	<u>63.00</u>
	Rethink-30	**31.54**	**51.50**	**68.00**

Table 1 presents the results. Even without rethinking, Rethink-1 already surpasses nearly all baselines, including models with up to $15\times$ more parameters. This demonstrates the learning efficiency gained by separating the latent thought vectors from the decoder: offloading reasoning structure to $\mathbf{z}$ frees the decoder from memorizing all patterns in its parameters, allowing a 0.2B model to outcompete much larger monolithic architectures. With 30 rethinking iterations (Rethink-30), performance further improves to 31.54% on GSM8K, 51.50% on SVAMP, and 68.00% on MultiArith, establishing new bests across all three benchmarks with the smallest model in the comparison.

Value of Rethinking. The gap between Rethink-1 and Rethink-30 quantifies the benefit of iterative refinement: 5.6 points on GSM8K, 4.1 on SVAMP, and 5.0 on MultiArith, all from the same model with additional test-time computation. This confirms that inference-time computation is an effective scaling axis: a smaller model that "thinks longer" surpasses a larger model that generates in one pass.

Out-of-Domain Generalization. The gains on SVAMP and MultiArith are particularly striking. Because the latent thought vector abstracts away surface-level token patterns, the model's reasoning strategy transfers across different problem wordings (SVAMP) and multi-step structures (MultiArith). Rethinking amplifies this effect: iterative optimization in latent space refines the underlying reasoning plan without overfitting to surface form, improving robustness to distribution shift.

4 Theoretical Understanding of Inference-Time Rethinking

The inference-time rethinking procedure described in Section 2.3 alternates between generating a reasoning trace and refining the latent thought vector. In this section, we show that this alternating procedure admits a principled variational interpretation: it performs coordinate-ascent optimization on a variational lower bound of $\log p_\theta(\mathbf{x}_q)$, equivalently minimizing the KL divergence between a factored variational distribution and the intractable joint posterior $p_\theta(\mathbf{x}_r, \mathbf{z}_0|\mathbf{x}_q)$.

4.1 VARIATIONAL FORMULATION

Given a question $\mathbf{x}_q$, our generative model defines a joint distribution over reasoning traces $\mathbf{x}_r$ and latent noise tokens $\mathbf{z}_0$:

$$p_\theta(\mathbf{x}_r, \mathbf{z}_0|\mathbf{x}_q) = \frac{p_\beta(\mathbf{x}_r|U_\alpha(\mathbf{z}_0), \mathbf{x}_q)\, p(\mathbf{z}_0)}{p_\theta(\mathbf{x}_q)}, \tag{8}$$

where $p(\mathbf{z}_0) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the base-space prior. Ideally, we would sample from this joint posterior to obtain both a good reasoning strategy ($\mathbf{z}_0$) and a corresponding trace ($\mathbf{x}_r$). However, the marginal $p_\theta(\mathbf{x}_q) = \int \sum_{\mathbf{x}_r} p_\beta(\mathbf{x}_r|U_\alpha(\mathbf{z}_0), \mathbf{x}_q)\, p(\mathbf{z}_0)\, d\mathbf{z}_0$ is intractable.

We introduce a **mean-field variational approximation** that factorizes over the two latent variables:

$$q(\mathbf{x}_r, \mathbf{z}_0) = q_1(\mathbf{x}_r)\, q_2(\mathbf{z}_0), \tag{9}$$

and seek to minimize the KL divergence to the true posterior:

$$\min_{q_1, q_2} D_{\mathrm{KL}}\big(q_1(\mathbf{x}_r)\, q_2(\mathbf{z}_0) \,\|\, p_\theta(\mathbf{x}_r, \mathbf{z}_0|\mathbf{x}_q)\big). \tag{10}$$

By the standard identity $\log p_\theta(\mathbf{x}_q) = \mathcal{L}(q_1, q_2) + D_{\mathrm{KL}}(q_1 q_2 \| p_\theta(\mathbf{x}_r, \mathbf{z}_0|\mathbf{x}_q))$, minimizing Eq. (10) is equivalent to maximizing the evidence lower bound (ELBO):

$$\mathcal{L}(q_1, q_2) = \mathbb{E}_{q_1(\mathbf{x}_r)q_2(\mathbf{z}_0)}\big[\log p_\beta(\mathbf{x}_r|U_\alpha(\mathbf{z}_0), \mathbf{x}_q)\big] - D_{\mathrm{KL}}\big(q_2(\mathbf{z}_0)\|p(\mathbf{z}_0)\big) + \mathcal{H}[q_1], \tag{11}$$

where $\mathcal{H}[q_1]$ denotes the entropy of q_1.

Intuition. The ELBO in Eq. (11) balances three objectives: (i) the expected log-likelihood encourages the latent thought and the trace to be mutually consistent (the thought should explain the trace, and the trace should be plausible under the thought); (ii) the KL term anchors the latent to the prior, preventing degenerate solutions; and (iii) the entropy term encourages exploration over traces, avoiding premature commitment to a single solution.

4.2 COORDINATE ASCENT RECOVERS THE RETHINKING PROCEDURE

We now show that the two steps of inference-time rethinking (Eqs. (6) and (7)) arise as coordinate-ascent updates on the ELBO in Eq. (11), under delta (point-mass) approximations.

Proposition 1 (Inference-Time Rethinking as Coordinate Ascent)**.** *Consider the variational objective in Eq. (10) with the mean-field factorization $q(\mathbf{x}_r, \mathbf{z}_0) = q_1(\mathbf{x}_r)\, q_2(\mathbf{z}_0)$. The optimal coordinate-ascent updates are:*

(i) ***Optimal q_1^* (fix q_2):*** *The optimal trace distribution satisfies*

$$\log q_1^*(\mathbf{x}_r) = \mathbb{E}_{q_2(\mathbf{z}_0)}\big[\log p_\beta(\mathbf{x}_r|U_\alpha(\mathbf{z}_0), \mathbf{x}_q)\big] + \text{const.} \tag{12}$$

When $q_2 = \delta(\mathbf{z}_0 - \hat{\mathbf{z}}_0)$ is a point mass, this reduces to the conditional:

$$q_1^*(\mathbf{x}_r) = p_\beta(\mathbf{x}_r|U_\alpha(\hat{\mathbf{z}}_0), \mathbf{x}_q), \tag{13}$$

which is realized by sampling $\mathbf{x}_r^{(t)} \sim p_\beta(\mathbf{x}_r|U_\alpha(\mathbf{z}_0^{(t-1)}), \mathbf{x}_q)$, i.e., the generation step (Eq. (6)).

(ii) ***Optimal q_2^* (fix q_1):*** *The optimal latent distribution satisfies*

$$\log q_2^*(\mathbf{z}_0) = \mathbb{E}_{q_1(\mathbf{x}_r)}\big[\log p_\beta(\mathbf{x}_r|U_\alpha(\mathbf{z}_0), \mathbf{x}_q)\big] + \log p(\mathbf{z}_0) + \text{const.} \tag{14}$$

When $q_1 = \delta(\mathbf{x}_r - \hat{\mathbf{x}}_r)$ is a point mass, this yields

$$q_2^*(\mathbf{z}_0) \propto p_\beta(\hat{\mathbf{x}}_r|U_\alpha(\mathbf{z}_0), \mathbf{x}_q)\, p(\mathbf{z}_0), \tag{15}$$

which is the true posterior $p_\theta(\mathbf{z}_0|\hat{\mathbf{x}}_r, \mathbf{x}_q)$. Approximating this posterior via ELBO maximization recovers the reflection step (Eq. (7)).

Proof. This follows from the standard mean-field coordinate-ascent derivation (Bishop, 2006; Blei et al., 2017). For a factored approximation $q(\mathbf{x}_r, \mathbf{z}_0) = q_1(\mathbf{x}_r)\, q_2(\mathbf{z}_0)$, the functional derivative of the ELBO with respect to q_1, holding q_2 fixed, yields the optimality condition:

$$\log q_1^*(\mathbf{x}_r) = \mathbb{E}_{q_2(\mathbf{z}_0)}\big[\log p_\theta(\mathbf{x}_r, \mathbf{z}_0|\mathbf{x}_q)\big] + \text{const.} \tag{16}$$

Expanding the joint using Eq. (8) and absorbing terms independent of $\mathbf{x}_r$ into the constant gives Eq. (12). Under the point-mass assumption $q_2 = \delta(\mathbf{z}_0 - \hat{\mathbf{z}}_0)$, the expectation collapses, yielding $q_1^*(\mathbf{x}_r) \propto p_\beta(\mathbf{x}_r|U_\alpha(\hat{\mathbf{z}}_0), \mathbf{x}_q)$, which is already normalized. The derivation for q_2^* is analogous: setting the functional derivative with respect to q_2 to zero, absorbing $\mathbf{x}_r$-independent terms, and substituting $q_1 = \delta(\mathbf{x}_r - \hat{\mathbf{x}}_r)$ gives Eq. (15). □

Intuition. The two steps have complementary roles. In Step (i), the model generates the best trace it can given its current understanding of the problem (encoded in $\hat{\mathbf{z}}_0$). This is analogous to a student writing out a solution based on their current plan. In Step (ii), the model examines the trace it just produced and asks: "what latent thought would best explain this reasoning?" By optimizing $\mathbf{z}_0$ to maximize the likelihood of the generated trace (regularized by the prior), the model revises its internal plan. This is analogous to the student re-reading their work and adjusting their strategy. The alternation allows errors in the trace to inform corrections in the thought, which then guides an improved trace in the next round.

4.3 Monotonic Convergence

A key consequence of the coordinate-ascent structure is the following convergence guarantee.

Proposition 2 (Monotonic ELBO Improvement)**.** *Let $\{(q_1^{(t)}, q_2^{(t)})\}_{t=0}^{T_{\text{rethink}}}$ be the sequence of variational distributions produced by the rethinking procedure. If each coordinate update does not decrease the ELBO, i.e., the q_1-update in Eq. (13) and the q_2-update in Eq. (15) each (approximately) maximize $\mathcal{L}$ with respect to their factor, then:*

$$\mathcal{L}(q_1^{(t+1)}, q_2^{(t+1)}) \geq \mathcal{L}(q_1^{(t)}, q_2^{(t)}) \quad \textit{for all } t \geq 0. \tag{17}$$

Equivalently, the KL divergence to the joint posterior is monotonically non-increasing:

$$D_{\mathrm{KL}}\big(q_1^{(t+1)} q_2^{(t+1)} \| p_\theta(\mathbf{x}_r, \mathbf{z}_0|\mathbf{x}_q)\big) \leq D_{\mathrm{KL}}\big(q_1^{(t)} q_2^{(t)} \| p_\theta(\mathbf{x}_r, \mathbf{z}_0|\mathbf{x}_q)\big). \tag{18}$$

Proof. Each coordinate update maximizes the ELBO over one factor while holding the other fixed. Since the ELBO is the negative of the KL (up to the constant $\log p_\theta(\mathbf{x}_q)$), and each update either increases or preserves the ELBO, the sequence $\{\mathcal{L}^{(t)}\}$ is non-decreasing. Since $\mathcal{L} \leq \log p_\theta(\mathbf{x}_q)$ is bounded above, the sequence converges. □

Intuition. This result provides a principled justification for why more rethinking iterations consistently help (Table 1): each round of generate-then-reflect is guaranteed to bring the variational approximation closer to (or at least no farther from) the true joint posterior over thoughts and traces. The model is not randomly searching; it is systematically descending a well-defined objective.

4.4 Connections and Remarks

Connection to Variational EM. The rethinking procedure can be viewed as a form of variational EM (Neal & Hinton, 1998) applied at test time with fixed model parameters θ. The generation step (updating q_1) corresponds to an "E-step" that imputes the missing trace, while the reflection step (updating q_2) corresponds to a second "E-step" that refines the latent variable. Both steps optimize the same ELBO, making this a pure inference procedure rather than a learning algorithm.

Connection to the Wake-Sleep Algorithm. There is a structural parallel with wake-sleep (Hinton et al., 1995): the generation step (sampling from the generative model) resembles the "sleep" phase, while the reflection step (inferring latent variables from observations) resembles the "wake" phase. However, a key difference is that our procedure optimizes a single consistent objective (the ELBO), whereas the classical sleep phase minimizes a reverse KL that is not jointly consistent with the wake objective.

Approximation Quality. Two sources of approximation merit discussion. First, the point-mass (delta) approximation for q_1 and q_2 sacrifices distributional uncertainty for computational efficiency; a richer family (e.g., Gaussian q_2, sampling multiple traces for q_1) would yield tighter bounds. Second, the reflection step uses finite gradient steps rather than exact optimization, so the per-step ELBO increase is approximate.

5 Discussion

Likelihood as a Proxy for Correctness. Our inference-time rethinking procedure refines the latent thought vector by maximizing the likelihood of the generated reasoning trace. This approach is effective because our model is trained on near-expert demonstrations: the learned distribution closely approximates correct reasoning, so the likelihood landscape naturally favors valid logical paths. A latent vector that produces a high-likelihood trace is statistically likely to encode a sound reasoning strategy. By iteratively optimizing the latent vector to better explain its own high-confidence generations, the model gravitates toward reasoning modes that are implicitly defined by the training data.

Generalizing Beyond Clean Supervision. This reliance on likelihood as a proxy for correctness has a clear limitation: it assumes high-quality training data. When the model is trained on uncurated or noisy supervision, the correlation between likelihood and correctness weakens, as high-likelihood traces may simply reflect common errors or misconceptions rather than valid reasoning. In such settings, relying solely on the generative prior and the decoder is insufficient for robust self-correction, and additional signals are needed to guide the rethinking process.

Towards Latent Planning and Verification. A natural next step is to incorporate explicit value signals into the rethinking loop. One direction is to introduce a *latent verifier* that predicts the correctness of a reasoning strategy directly from the thought vector, before any tokens are decoded. This would transform rethinking into a form of latent planning (Kong et al., 2024a;b; Qin et al., 2025), where the model evaluates and selects among candidate strategies in the continuous latent space. Another promising direction is to leverage discrete external rewards, such as feedback from a code compiler or a symbolic checker, to perform *test-time policy gradient optimization* (Li et al., 2025) over the latent thought vector. Regularizing toward the learned prior naturally serves as a trust region (Schulman et al., 2017; Noh et al., 2025), keeping the optimization within the manifold where the decoder remains well-calibrated and bridging intuitive generation with deliberative, reward-guided search.

6 Conclusion

We introduce *Inference-Time Rethinking*, a framework that decouples reasoning into latent thought vectors and procedural generation, and iteratively refines the thought vector at test time via a Gibbs-style loop. Offloading reasoning structure to latent variables enables efficient training: a 0.2B model already surpasses much larger baselines in a single pass, while the continuous, training-shaped latent space supports test-time gradient-based self-correction through additional compute. On GSM8K, SVAMP, and MultiArith, our model establishes new bests while remaining the smallest in the comparison, demonstrating that inference-time computation over a well-structured latent space is a powerful scaling axis complementary to parameter count. In this work we focus exclusively on mathematical reasoning, which recent evidence suggests can bootstrap broader reasoning capabilities across domains (Pang et al., 2025). Nevertheless, extending our framework to other forms of structured reasoning, such as deductive, inductive, and transductive inference, remains an important direction for future work.

Acknowledgments

Y. W. is partially supported by NSF DMS-2415226, DARPA W912CG25CA007 and research gift funds from Amazon and Qualcomm. We thank Luca Zancato and Ben Bowman of Amazon AWS Research for their valuable help and discussions. We gratefully acknowledge the support of Lambda, Inc. for providing the computational resources used in this project.

References

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, 2016.

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

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*, 2023.

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

Angela D Friederici. The brain basis of language processing: from structure to function. *Physiological reviews*, 91(4):1357–1392, 2011.

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

Geoffrey E. Hinton, Peter Dayan, Brendan J. Frey, and Radford M. Neal. The wake-sleep algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. Mawps: A math word problem repository. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pp. 1152–1157, 2016.

Deqian Kong, Yuhao Huang, Jianwen Xie, Edouardo Honig, Ming Xu, Shuanghong Xue, Pei Lin, Sanping Zhou, Sheng Zhong, Nanning Zheng, et al. Molecule design by latent prompt transformer. *Advances in Neural Information Processing Systems*, 37:89069–89097, 2024a.

Deqian Kong, Dehong Xu, Minglu Zhao, Bo Pang, Jianwen Xie, Andrew Lizarraga, Yuhao Huang, Sirui Xie, and Ying Nian Wu. Latent plan transformer for trajectory abstraction: Planning as latent space inference. *Advances in Neural Information Processing Systems*, 37:123379–123401, 2024b.

Deqian Kong, Minglu Zhao, Dehong Xu, Bo Pang, Shu Wang, Edouardo Honig, Zhangzhang Si, Chuan Li, Jianwen Xie, Sirui Xie, and Ying Nian Wu. Latent thought models with variational bayes inference-time computation. In *42nd International Conference on Machine Learning (ICML)*, 2025.

Hengli Li, Chenxi Li, Tong Wu, Xuekai Zhu, Yuxuan Wang, Zhaoxin Yu, Eric Hanchen Jiang, Song-Chun Zhu, Zixia Jia, Ying Nian Wu, and Zilong Zheng. Seek in the dark: Reasoning via test-time instance-level policy gradient in latent space. *arXiv preprint arXiv:2505.13308*, 2025.

Jiayu Liu, Zhenya Huang, Anya Sims, Enhong Chen, Yee Whye Teh, and Ning Miao. Marcos: Deep thinking by markov chain of continuous thoughts. *arXiv preprint arXiv:2509.25020*, 2025.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

Radford M. Neal and Geoffrey E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In Michael I. Jordan (ed.), *Learning in Graphical Models*, pp. 355–368. Kluwer Academic Publishers, 1998.

Donghun Noh, Deqian Kong, Minglu Zhao, Andrew Lizarraga, Jianwen Xie, Ying Nian Wu, and Dennis Hong. Latent adaptive planner for dynamic manipulation. In *Proceedings of the 9th Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 2430–2448, 2025.

Bo Pang, Deqian Kong, Silvio Savarese, Caiming Xiong, and Yingbo Zhou. Reasoning curriculum: Bootstrapping broad llm reasoning from math. *arXiv preprint arXiv:2510.26143*, 2025.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*, 2021.

Aoyang Qin, Deqian Kong, Wei Wang, Ying Nian Wu, Song-Chun Zhu, and Sirui Xie. Generative actor critic. *arXiv preprint arXiv:2512.21527*, 2025.

Subhro Roy and Dan Roth. Solving general arithmetic word problems. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1743–1752, 2015.

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

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

Wenhui Tan, Jiaze Li, Jianzhong Ju, Zhenbo Luo, Jian Luan, and Ruihua Song. Think silently, think fast: Dynamic latent compression of llm reasoning chains. *arXiv preprint arXiv:2505.16552*, 2025.

Qwen Team et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3), 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Michael T Ullman. Contributions of memory circuits to language: The declarative/procedural model. *Cognition*, 92(1-2):231–270, 2004.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

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