

Latent Thoughts, Certain Gains: Bayesian Optimization for LLM Reasoning

Xin Xiao and Jiang Zhong*

School of Computer Science, Chongqing University
{20241401023@stu.cqu.edu.cn, zhongjiang@cqu.edu.cn}

Abstract

Recent advances in Large Language Models (LLMs) have shifted toward latent reasoning, where intermediate processes are represented as continuous vectors rather than text tokens. While this paradigm reduces inference cost, existing approaches often suffer from instability on complex, out-of-distribution tasks where robust reasoning is critical. To address this, we propose Bayesian Latent Thought Optimization (BLTO), a training-free framework that enhances frozen LLMs entirely at test time. BLTO treats latent thought tokens as stochastic variables and formulates reasoning as a Bayesian inference problem. We introduce a variational posterior over latent vectors and optimize it by maximizing a reward-weighted objective that balances reasoning quality with KL regularization toward a standard Gaussian prior. The reward is derived from the model’s predictive consistency via output entropy, enabling fully self-supervised optimization without external supervision. BLTO employs reparameterized sampling and reward-weighted updates within a variational inference framework, together with an adaptive controller that dynamically determines inference steps based on reward improvement and uncertainty convergence. Extensive experiments on challenging reasoning benchmarks demonstrate that BLTO consistently improves frozen LLM performance.

1 Introduction

Recent advances in large language models (LLMs) (Yang et al., 2025; Guo et al., 2025; Grattafiori et al., 2024) have significantly improved the ability of neural systems to perform complex reasoning tasks, including mathematical problem solving, symbolic manipulation, and multi-step logical inference (Ahn et al., 2024; Patil and Jadon, 2025). A major driver of this progress is the emergence

of Chain-of-Thought (CoT) prompting (Wei et al., 2022), which encourages models to generate intermediate reasoning steps before producing final answers. By externalizing the reasoning process into natural language, CoT allows LLMs to decompose complex problems into manageable substeps, leading to substantial performance gains across a wide range of reasoning benchmarks.

Despite its effectiveness, explicit CoT reasoning introduces several practical limitations. Generating intermediate reasoning tokens significantly increases inference cost, latency, and memory consumption (Feng et al., 2025), particularly for long reasoning chains. More importantly, textual reasoning traces may not faithfully reflect the model’s internal reasoning dynamics and can sometimes introduce spurious or redundant steps (Lee and Hockenmaier, 2025). These issues have motivated a growing line of research on latent reasoning (Chen et al., 2025; Wang et al.), where intermediate reasoning processes are represented as continuous hidden states rather than explicit textual tokens. By operating directly in the embedding space, latent reasoning methods can substantially reduce token overhead while preserving the model’s capacity for multi-step inference (Sun et al., 2025). For instance, Coconut (Hao et al., 2024) first proposes the method of “thinking” in a continuous Latent Space, but its performance is poor. SoftCoT (Xu et al., 2025b) utilizes a trained small LLM as an auxiliary model to generate Latent Thought.

However, existing latent reasoning approaches often suffer from limited robustness and instability, particularly on challenging or out-of-distribution reasoning tasks. Because latent reasoning operates in a high-dimensional continuous space without explicit structural constraints, the model may struggle to discover effective intermediate reasoning representations during inference. As a result, many existing approaches rely on fixed latent prompts or heuristic search strategies that provide limited

*Corresponding author.

adaptability to individual problem instances.

In this work, we propose Bayesian Latent Thought Optimization (BLTO), a framework that enhances reasoning in frozen LLMs via test-time optimization in latent space. BLTO models latent thoughts as stochastic variables and formulates reasoning as Bayesian inference over cognitive states. It optimizes a variational posterior using reward-weighted objectives that balance quality and prior regularization, with rewards derived from the model’s own output entropy—enabling fully self-supervised optimization. To enable efficient exploration, BLTO combines reparameterized sampling with reward-weighted updates and an adaptive inference controller that dynamically adjusts optimization steps based on reward improvement. Experimental results on several challenging reasoning benchmarks demonstrate that BLTO consistently improves the performance of frozen LLMs across tasks, highlighting its ability to recover effective reasoning representations in challenging settings. Our contributions can be summarized as follows:

(1) We introduce Bayesian Latent Thought Optimization (BLTO), a training-free framework that formulates latent reasoning in frozen LLMs as a Bayesian optimization problem in latent embedding space, enabling principled inference-time improvement of reasoning capabilities.

(2) We develop a variational inference approach with reward-weighted updates and an adaptive optimization controller, allowing self-supervised search for high-quality latent thought representations without external supervision or explicit reasoning tokens.

(3) We conduct extensive experiments on multiple challenging reasoning benchmarks, including mathematical and symbolic reasoning tasks, demonstrating that BLTO significantly outperforms existing latent reasoning methods.

2 Related Work

2.1 From Explicit to Latent Reasoning

The pursuit of advanced reasoning in LLMs has evolved from explicit textual rationales to efficient latent representations. Chain-of-Thought (CoT) prompting (Wei et al., 2022; Kojima et al., 2022) established that intermediate reasoning steps improve performance, but incur high latency and token costs. Latent reasoning addresses this by operating in continuous space. Existing Latent

Thought methods fall into two categories: fine-tuning-based and auxiliary module-based. Fine-tuning methods modify model parameters: iCoT (Deng et al., 2024) enables reasoning and answer generation in a single forward pass; Coconut (Hao et al., 2024) uses last-layer hidden states as "continuous thoughts"; CODI (Shen et al., 2025) employs self-distillation for implicit reasoning; and CCOT (Cheng and Van Durme, 2024) compresses reasoning into "meditation" tokens. Auxiliary module methods preserve the base LLM frozen: SoftCoT (Xu et al., 2025b) trains a lightweight model to generate "soft thought" tokens projected into the embedding space. LTPO (Ye et al., 2025) further optimizes latent vectors during inference via policy gradients, using the model’s own confidence as reward without parameter updates.

2.2 Reinforcement Learning for LLM Optimization

A parallel line of research applies Reinforcement Learning (RL) to enhance LLM capabilities. The dominant paradigm, originating with Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022), involves fine-tuning model parameters using algorithms such as PPO (Schulman et al., 2017) to align outputs with human preferences. This approach has been extended by more efficient variants like DPO (Rafailov et al., 2023), which bypasses explicit reward modeling, and specialized techniques for mathematical reasoning including GRPO (Shao et al., 2024) and DAPO (Yu et al., 2025). Despite their diversity, these methods universally operate by updating model weights during an offline training stage, guided by external reward signals derived from human feedback or verifiable ground-truth labels. More recently, a growing interest has emerged in test-time optimization strategies that adapt model behavior at inference without parameter updates (Li et al., 2025). However, existing approaches either require external verifiers or rely on expensive text generation within the optimization loop, limiting their practical applicability.

3 Methodology

We propose Bayesian Latent Thought Optimization (BLTO), a training-free framework that enhances LLM reasoning by formulating test-time latent thought optimization as a Bayesian inference problem, as shown in Figure 1. BLTO treats inter-

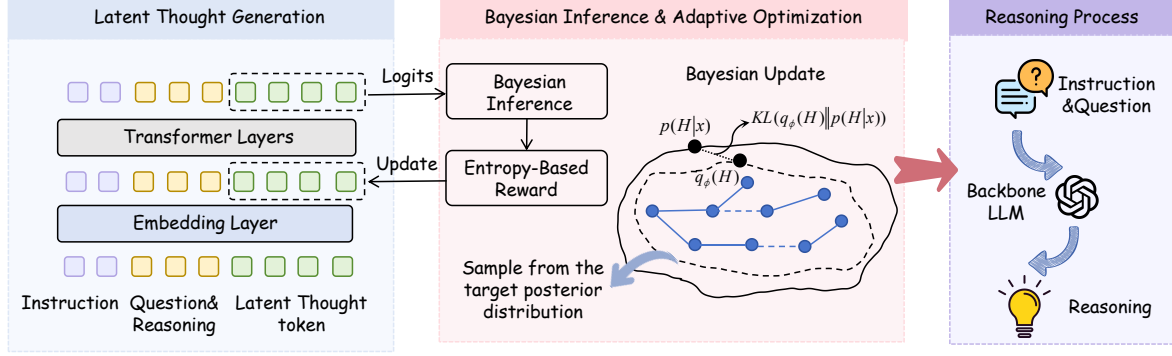


Figure 1: Illustration of Bayesian Latent Thought Optimization. BLTO models latent reasoning paths as stochastic variables. A variational posterior is learned online for each instance by maximizing an intrinsic reward, regularized by KL divergence to a prior distribution. This allows the frozen LLM to explore high-confidence reasoning trajectories in the continuous latent space during inference.

mediate latent thought vectors as stochastic variables, introducing a variational posterior over the latent space that is iteratively refined to maximize a confidence-based reward signal derived from the frozen model’s own output distributions. This approach enables principled exploration of the latent reasoning space while maintaining distributional stability through KL regularization toward a standard Gaussian prior.

3.1 Latent Thought Representation

Let M_θ denote a frozen auto-regressive language model with parameters θ . Given an input prompt consisting of discrete tokens $\mathbf{x} = (x_1, \dots, x_N)$, the model’s embedding layer $\mathcal{E}$ maps it to a sequence of continuous vectors $\mathcal{E}(\mathbf{x}) = (\mathbf{e}_1, \dots, \mathbf{e}_N) \in \mathbb{R}^{N \times d}$, where d is the hidden dimension.

To enable latent reasoning, we augment the input with K special placeholder tokens (e.g., [THINK]), whose embeddings $\mathbf{H} = (\mathbf{h}_1, \dots, \mathbf{h}_K) \in \mathbb{R}^{K \times d}$ serve as learnable latent thought vectors. These tokens represent abstract, continuous reasoning states that can be optimized to improve the model’s subsequent generation. The complete input to the model is the concatenation $\mathcal{E}(\mathbf{x}) \parallel \mathbf{H}$.

We adopt a Bayesian perspective on latent thought optimization. Instead of seeking a single optimal $\mathbf{H}$, we aim to infer the posterior distribution over latent thoughts conditioned on the prompt and a task-specific reward signal:

$$p(\mathbf{H}|\mathbf{x}) \propto p(\mathbf{H}) \cdot \exp(\beta \cdot \mathcal{R}(\mathbf{x}, \mathbf{H})), \quad (1)$$

where $p(\mathbf{H})$ is a prior distribution over latent thoughts, $\mathcal{R}(\mathbf{x}, \mathbf{H})$ is a reward function measuring the quality of latent thoughts, and $\beta > 0$ is

an inverse temperature parameter controlling the sharpness of the posterior.

3.2 Bayesian Inference for Latent Thoughts

Direct inference of the posterior in Eq. (1) is intractable due to the complexity of the reward function and the high-dimensional continuous space $\mathbb{R}^{K \times d}$. The reward $\mathcal{R}(\mathbf{x}, \mathbf{H})$ is non-linear and non-convex, and the latent dimensionality ($K \times d$, typically thousands) precludes exhaustive sampling methods.

We employ variational inference to approximate the intractable posterior. Let $\mathcal{Q}$ be a family of distributions over $\mathbf{H}$ parameterized by ϕ . We seek $q_\phi(\mathbf{H}) \in \mathcal{Q}$ that minimizes the KL divergence to the true posterior:

$$q_{\phi^*}(\mathbf{H}) = \arg \min_{q_\phi \in \mathcal{Q}} \text{KL}(q_\phi(\mathbf{H}) \| p(\mathbf{H}|\mathbf{x})). \quad (2)$$

Expanding the KL divergence reveals its relationship to the evidence lower bound (ELBO):

$$\begin{aligned} \text{KL}(q_\phi(\mathbf{H}) \| p(\mathbf{H}|\mathbf{x})) &= \mathbb{E}_{q_\phi} \left[\log \frac{q_\phi(\mathbf{H})}{p(\mathbf{H}|\mathbf{x})} \right] \\ &= \mathbb{E}_{q_\phi} [\log q_\phi(\mathbf{H}) - \log p(\mathbf{H}|\mathbf{x})] \\ &= \mathbb{E}_{q_\phi} \left[\log q_\phi(\mathbf{H}) - \log \frac{p(\mathbf{x}|\mathbf{H})p(\mathbf{H})}{p(\mathbf{x})} \right] \\ &= \mathbb{E}_{q_\phi} [\log q_\phi(\mathbf{H}) - \log p(\mathbf{x}|\mathbf{H}) \\ &\quad - \log p(\mathbf{H})] + \log p(\mathbf{x}) \\ &= -(\mathbb{E}_{q_\phi} [\log p(\mathbf{x}|\mathbf{H})] \\ &\quad - \text{KL}(q_\phi(\mathbf{H}) \| p(\mathbf{H}))) + \log p(\mathbf{x}). \end{aligned} \quad (3)$$

Since $\log p(\mathbf{x})$ is constant w.r.t. ϕ , minimizing KL divergence is equivalent to maximizing the

ELBO:

$$\text{ELBO}(\phi) = \mathbb{E}_{q_\phi} [\log p(\mathbf{x}|\mathbf{H})] - \text{KL}(q_\phi(\mathbf{H})\|p(\mathbf{H})). \quad (4)$$

From Eq. (1), we have $\log p(\mathbf{x}|\mathbf{H}) \propto \beta \cdot \mathcal{R}(\mathbf{x}, \mathbf{H})$. Substituting into the ELBO yields our minimization objective:

$$\mathcal{L}(\phi) = \text{KL}(q_\phi(\mathbf{H})\|p(\mathbf{H})) - \beta \cdot \mathbb{E}_{q_\phi} [\mathcal{R}(\mathbf{x}, \mathbf{H})]. \quad (5)$$

This objective balances two terms: the KL divergence regularizes the variational distribution toward the prior, while the expected reward drives it toward regions of high confidence.

3.3 Variational Approximation

To make variational inference tractable, we specify a concrete variational family Q . This choice involves a trade-off between expressiveness and computational efficiency. While more expressive families (e.g., normalizing flows, full-covariance Gaussians) could capture complex posterior geometries, they introduce significant computational overhead. We therefore adopt a factorized Gaussian variational distribution with diagonal covariance, which strikes an effective balance between flexibility and tractability.

Let $\phi = \{\boldsymbol{\mu}, \boldsymbol{\rho}\}$ denote the variational parameters, where $\boldsymbol{\mu} \in \mathbb{R}^{K \times d}$ is the mean and $\boldsymbol{\rho} \in \mathbb{R}^{K \times d}$ is the log-variance. The variational posterior factorizes across latent tokens:

$$q_\phi(\mathbf{H}) = \prod_{k=1}^K \mathcal{N}(\mathbf{h}_k; \boldsymbol{\mu}_k, \text{diag}(\boldsymbol{\sigma}_k^2)), \quad \boldsymbol{\sigma}_k^2 = \exp(\boldsymbol{\rho}_k). \quad (6)$$

The use of log-variances ensures $\boldsymbol{\sigma}_k^2$ remains positive without constrained optimization, as $\exp(\cdot)$ maps the entire real line to positive reals.

For the prior distribution $p(\mathbf{H})$, we adopt a standard isotropic Gaussian:

$$p(\mathbf{H}) = \prod_{k=1}^K \mathcal{N}(\mathbf{h}_k; \mathbf{0}, \mathbf{I}) = (2\pi)^{-\frac{Kd}{2}} \exp\left(-\frac{1}{2} \sum_{k=1}^K \|\mathbf{h}_k\|^2\right). \quad (7)$$

This choice encodes a preference for latent thoughts with small norms, acting as a natural regularizer that prevents optimization from drifting too far from the pretrained embedding space. In the absence of strong reward signals, the posterior reverts

to the prior, corresponding to minimal intervention in the model’s native behavior.

A key advantage of using Gaussian distributions for both the variational posterior and the prior is that the KL divergence term has a closed-form expression. For two multivariate Gaussian distributions $q(\mathbf{h}) = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ and $p(\mathbf{h}) = \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ with diagonal covariances, the KL divergence is:

$$\begin{aligned} \text{KL}(q\|p) = & \frac{1}{2} \left[\text{tr}(\boldsymbol{\Sigma}_p^{-1} \boldsymbol{\Sigma}_q) \right. \\ & + (\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)^\top \boldsymbol{\Sigma}_p^{-1} (\boldsymbol{\mu}_p - \boldsymbol{\mu}_q) \\ & \left. - d + \log \frac{\det \boldsymbol{\Sigma}_p}{\det \boldsymbol{\Sigma}_q} \right]. \end{aligned} \quad (8)$$

With our choice of prior $\boldsymbol{\mu}_p = \mathbf{0}$, $\boldsymbol{\Sigma}_p = \mathbf{I}$, and variational posterior $\boldsymbol{\Sigma}_q = \text{diag}(\boldsymbol{\sigma}^2)$, this simplifies to:

$$\begin{aligned} \text{KL}(q_\phi(\mathbf{H})\|p(\mathbf{H})) = & \frac{1}{2} \sum_{k=1}^K \sum_{j=1}^d \left(\sigma_{k,j}^2 + \mu_{k,j}^2 \right. \\ & \left. - 1 - \log \sigma_{k,j}^2 \right). \end{aligned} \quad (9)$$

This closed-form expression has several important properties: it is differentiable, provides an analytical regularizer that reduces gradient variance, and has an intuitive interpretation—the term $\sigma_{k,j}^2 - \log \sigma_{k,j}^2 - 1$ encourages the variational variance to match the prior variance of 1, while $\mu_{k,j}^2$ penalizes deviations of the mean from zero.

We initialize the mean parameters using the embeddings of the placeholder tokens with small Gaussian noise:

$$\boldsymbol{\mu}^{(0)} = \mathcal{E}([\text{THINK}]_1, \dots, [\text{THINK}]_K) + \boldsymbol{\delta}, \quad (10)$$

where $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \epsilon^2 \mathbf{I})$. This breaks symmetry and encourages diverse exploration across tokens. The log-variances are initialized as $\boldsymbol{\rho}^{(0)} = \log(\sigma_0^2) \mathbf{1}$, corresponding to an initial standard deviation that allows sufficient exploration while maintaining stability. As optimization progresses, the variance may adapt, decreasing in confident regions or increasing where multiple solutions are equally plausible.

3.4 Entropy-Based Intrinsic Reward

To guide latent thought optimization without external supervision, we design an intrinsic reward derived from the entropy of the frozen LLM’s output distributions. Intuitively, coherent latent reasoning should lead to lower predictive entropy.

Given latent thoughts $\mathbf{H}$, we perform a forward pass through the frozen model:

$$(\mathbf{P}_1, \dots, \mathbf{P}_{N+K}) = \text{softmax}(M_\theta(\mathcal{E}(\mathbf{x}) \parallel \mathbf{H})), \quad (11)$$

where $\mathbf{P}_i$ denotes the vocabulary distribution at position i and positions $N+1, \dots, N+K$ correspond to latent tokens.

We define the entropy of the k -th latent token as

$$\mathcal{H}_k = - \sum_{v \in \mathcal{V}} \mathbf{P}_{N+k}(v) \log \mathbf{P}_{N+k}(v), \quad (12)$$

and use the negative average entropy as the intrinsic reward:

$$\mathcal{R}(\mathbf{x}, \mathbf{H}) = -\frac{1}{K} \sum_{k=1}^K \mathcal{H}_k. \quad (13)$$

The reward is fully differentiable with respect to $\mathbf{H}$ through backpropagation. For numerical stability, we compute entropy as $-\sum_v P \log(P + \varepsilon)$ with $\varepsilon = 10^{-10}$ and clip log-probabilities to $[-20, 0]$.

We optimize the variational parameters $\phi = \{\boldsymbol{\mu}, \boldsymbol{\rho}\}$ by minimizing

$$\mathcal{L}(\phi) = \text{KL}(q_\phi(\mathbf{H}) \parallel p(\mathbf{H})) - \beta \mathbb{E}_{q_\phi}[\mathcal{R}(\mathbf{x}, \mathbf{H})]. \quad (14)$$

Gradients are estimated using the reparameterization trick $\mathbf{H} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{\sigma} = \exp(\frac{1}{2}\boldsymbol{\rho})$. Parameters are updated with AdamW and $\boldsymbol{\rho}$ is clipped to $[-14, -2.3]$ for stability.

We terminate optimization when either the mean posterior uncertainty $\bar{\sigma}_t = \frac{1}{Kd} \sum_{k,j} \sigma_{k,j}^{(t)}$ falls below a threshold ϵ_{unc} , or the moving-average reward improvement satisfies $|\bar{\mathcal{R}}_t - \bar{\mathcal{R}}_{t-\tau}| < \epsilon_{\text{rew}}$. This adaptive stopping mechanism allocates more optimization steps to difficult inputs while preventing over-optimization.

3.5 Optimization and Adaptive Inference

We optimize the variational parameters $\phi = \{\boldsymbol{\mu}, \boldsymbol{\rho}\}$ by minimizing the objective $\mathcal{L}(\phi) = \text{KL}(q_\phi \parallel p) - \beta \mathbb{E}_{q_\phi}[\mathcal{R}]$. Computing gradients of the expectation term with standard score-function estimators suffers from high variance, which can destabilize optimization. We instead employ the reparameterization trick to obtain low-variance gradient estimates.

Using the reparameterization $\mathbf{H} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{\sigma} = \exp(\frac{1}{2}\boldsymbol{\rho})$, the expecta-

tion becomes $\mathbb{E}_{q_\phi}[\mathcal{R}] = \mathbb{E}_\epsilon[\mathcal{R}(\mathbf{x}, \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon})]$. Approximating with S Monte Carlo samples yields:

$$\nabla_{\boldsymbol{\mu}} \mathcal{L} = \boldsymbol{\mu} - \frac{\beta}{S} \sum_{s=1}^S \nabla_{\mathbf{H}} \mathcal{R}^{(s)}, \quad (15)$$

$$\begin{aligned} \nabla_{\boldsymbol{\rho}} \mathcal{L} = & \frac{1}{2}(\boldsymbol{\sigma}^2 - \mathbf{1}) \\ & - \frac{\beta}{2S} \sum_{s=1}^S (\nabla_{\mathbf{H}} \mathcal{R}^{(s)} \odot \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}^{(s)}), \end{aligned} \quad (16)$$

where $\mathcal{R}^{(s)} = \mathcal{R}(\mathbf{x}, \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}^{(s)})$. These gradient expressions have intuitive interpretations: the mean gradient balances prior regularization with reward maximization, while the log-variance gradient automatically adjusts exploration based on both uncertainty and reward landscape curvature. Parameters are updated via AdamW with learning rate η , and we clip $\boldsymbol{\rho}$ to $[-14, -2.3]$ for numerical stability.

A key advantage of our Bayesian formulation is principled adaptive inference using posterior uncertainty. Let $\bar{\mathcal{R}}_t$ be the exponential moving average of rewards with decay $\alpha \in [0.9, 0.99]$, and $\bar{\sigma}_t = \frac{1}{Kd} \sum_{k,j} \sigma_{k,j}^{(t)}$ the average standard deviation. We define three convergence criteria: reward stagnation over a window of τ steps, uncertainty collapse below ϵ_{unc} , and KL stabilization between successive variational distributions. The complete stopping condition is:

$$\begin{aligned} t \geq T_{\min} \quad \wedge \quad & (t \geq T_{\max} \quad \vee \quad \bar{\sigma}_t < \epsilon_{\text{unc}} \\ & \vee \quad \max_{i=0}^{\tau-1} |\bar{\mathcal{R}}_{t-i} - \bar{\mathcal{R}}_{t-i-\tau}| < \epsilon_{\text{rew}}). \end{aligned} \quad (17)$$

This multi-criteria approach automatically allocates more steps to complex problems while avoiding over-optimization on simple instances.

We further adapt the exploration-exploitation trade-off using uncertainty estimates. We introduce an adaptive temperature parameter λ_t that scales the sampling noise:

$$\begin{aligned} \mathbf{H}^{(s)} = & \boldsymbol{\mu} + \lambda_t \cdot \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}^{(s)}, \\ \lambda_t = & \lambda_{\min} + (\lambda_{\max} - \lambda_{\min}) \cdot \frac{\bar{\sigma}_t}{\bar{\sigma}_0}, \end{aligned} \quad (18)$$

where $\bar{\sigma}_0$ is the initial average standard deviation. This schedule implements a form of simulated annealing: early in optimization when uncertainty is high, $\lambda_t \approx \lambda_{\max}$ promotes broad exploration; as uncertainty decreases and the posterior collapses, λ_t approaches $\lambda_{\min}$, focusing on exploitation around the current mean.

Table 1: Performance of BLTO vs. baselines across LLaMA-3.1-8B-Instruct and Qwen-2.5-7B-Instruct. Accuracy (%) is reported. The optimal results are in bold.

Model & Method	GSM8K	MATH500	ASDiv-Aug	AIME2024	AIME2025	Avg.
<i>LLaMA-3.1-8B-Instruct</i>						
Zero-Shot CoT (Kojima et al., 2022)	76.88	48.60	79.58	6.67	0.00	42.35
Zero-Shot CoT-Unk (Kojima et al., 2022)	80.67	46.20	89.50	10.00	3.33	45.94
SoftCoT (Xu et al., 2025b)	80.36	39.80	87.57	0.00	0.00	41.55
LTPO (Ye et al., 2025)	80.89	46.60	88.25	10.00	6.67	46.48
Ours	82.03	51.00	90.08	16.67	6.67	49.29
<i>Qwen-2.5-7B-Instruct</i>						
Zero-Shot CoT (Kojima et al., 2022)	88.55	72.40	91.71	10.00	10.00	54.53
Zero-Shot CoT-Unk (Kojima et al., 2022)	86.20	70.80	91.52	13.33	10.00	54.37
SoftCoT (Xu et al., 2025b)	85.29	65.20	88.92	0.00	0.00	47.88
LTPO (Ye et al., 2025)	87.87	71.80	91.71	10.00	3.33	52.94
Ours	88.55	74.80	92.58	20.00	16.67	58.52

4 Experiments

4.1 Experimental Setup

Models. We evaluate our method on open-source instruction-tuned language models from two distinct families: LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024) and Qwen-2.5-7B-Instruct (Qwen et al., 2025). These models are fully open-source, ensuring reproducibility, and span different architectural designs and training paradigms.

Datasets. Following recent work (Deng et al., 2024; Ye et al., 2025), we evaluate on five mathematical reasoning datasets: GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021), ASDiv (Miao et al., 2020), AIME2024¹, and AIME2025². GSM8K and MATH-500 serve as standard benchmarks for direct comparison with prior methods. ASDiv comprises elementary math word problems with varied linguistic patterns. The AIME datasets feature challenging competition problems that have become standard testbeds for evaluating state-of-the-art reasoning models. For ASDiv, we adopt the augmented version (ASDiv-Aug) (Xu et al., 2025b) to ensure test instances are novel and prevent potential data contamination, given that LLaMA and Qwen models are extensively pretrained on large-scale web corpora.

Baselines. We compare our method against several strong baselines: (1) **Zero-Shot Chain-of-Thought (Zero-Shot CoT)** (Kojima et al., 2022) serves as our primary baseline, eliciting reasoning

by instructing the model to generate explicit step-by-step thinking in discrete token space. (2) **Zero-Shot CoT with Untuned Tokens (Zero-Shot CoT-Unk)** augments the prompt with randomly initialized special tokens that remain fixed during inference, controlling for the presence of additional tokens without adaptive refinement. (3) **SoftCoT** (Xu et al., 2025b) performs reasoning in continuous latent space by optimizing soft prompts at test time, demonstrating superior performance compared to PEFT methods like LoRA (Hu et al., 2022) and latent reasoning approaches such as Coconut (Hao et al., 2024). (4) **LTPO** (Ye et al., 2025): The previous state-of-the-art in test-time latent thought optimization, which employs policy gradients with a confidence-based intrinsic reward derived from the frozen LLM’s output distributions.

Implementation Details. All experiments are conducted on a single NVIDIA A100 GPU. We sample 5 latent trajectories per step, with $\beta = 0.8$ and KL weight 0.008. Adaptive steps range from 5 to 22 with patience 5, improvement threshold 0.006, and early stopping when posterior uncertainty drops below 0.025 after 3 warmup steps. Generation uses top- $k = 10$ with 4096 max tokens.

4.2 Main Results

Overall Performance. Table 1 presents the main results comparing BLTO against all baselines across two model families and five datasets. BLTO consistently outperforms all baselines on both models, achieving average accuracy improvements of 2.81% over LTPO on LLaMA-3.1-8B and 5.58% over LTPO on Qwen-2.5-7B. These results demonstrate that Bayesian latent thought optimization ef-

¹https://huggingface.co/datasets/Maxwell-Jia/AIME_2024

²<https://huggingface.co/datasets/opencompass/AIME2025>

Table 2: Comparison with training-based methods on GSM8K, MATH-500, and AIME2024 using LLaMA-3.1-8B-Instruct. “Train Params” indicates whether model weights are updated. † denotes results from (Zeng et al., 2025). Best results in bold, second-best underlined.

Method	Train Model Params	Supervision Type	GSM8K	MATH 500	AIME 2024
Genius	Yes	Self-supervised	78.09	47.60	3.33
SimpleRL-Zoo†	Yes	External Data	79.20	23.00	0.00
iCoT	Yes	External Data	50.05	-	-
SoftCoT	Yes	External Data	80.36	39.80	0.00
LTPO	No	Self-supervised	80.89	46.60	10.00
BLTO (Ours)	No	Self-supervised	82.03	51.00	16.67

Table 3: Performance on commonsense and symbolic reasoning tasks. Accuracy (%) reported on StrategyQA and Symbolic Date Understanding (DU).

Method	StrategyQA	Symbolic DU
<i>LLaMA-3.1-8B-Instruct</i>		
Zero-Shot CoT	71.49	55.6
Zero-Shot CoT-Unk	69.67	57.6
SoftCoT	67.34	58.4
LTPO	72.18	62.0
BLTO (Ours)	71.72	65.2
<i>Qwen2.5-7B-Instruct</i>		
Zero-Shot CoT	72.63	64.4
Zero-Shot CoT-Unk	71.95	75.2
SoftCoT	62.75	68.4
LTPO	71.80	76.4
BLTO (Ours)	73.69	76.8

fectively handles complex mathematical reasoning tasks where multi-step inference is critical.

Difficulty-Dependent Gains. Analysis across datasets shows that BLTO’s performance gains increase with task difficulty. On relatively straightforward benchmarks such as GSM8K and ASDiv-Aug, improvements over LTPO are modest, ranging from one to two percent. In contrast, on more challenging datasets like MATH-500, gains reach three to four percent, and on AIME, improvements are particularly pronounced, ranging from 6.67 to 10 percent. This trend is consistent with our adaptive mechanism, where more difficult problems necessitate extensive exploration of the latent space guided by Bayesian uncertainty estimates.

Comparison with Training-Based Methods Table 2 compares BLTO against training-based methods that update model parameters. Remarkably, BLTO achieves the highest accuracy across all three datasets without any parameter updates. On GSM8K, BLTO reaches 82.03%, surpassing SoftCoT at 80.36% and LTPO at 80.89%. On MATH-

Table 4: Comparison of LTPO and Our Method on Mathematical Reasoning Datasets. We report the accuracy (%) of different Qwen2.5 base models on the GSM8K and ASDiv-Aug benchmarks.

Dataset	Method	0.5B	1.5B	7B
GSM8K	LTPO	13.57	40.80	87.87
	Ours	37.76	57.86	88.55
ASDiv-Aug	LTPO	50.87	56.07	91.71
	Ours	75.53	78.90	92.58

500, BLTO achieves 51.00%, substantially outperforming Genius (Xu et al., 2025a) at 47.60%. The most striking gains appear on AIME2024, where BLTO attains 16.67% accuracy. These results demonstrate that test-time optimization in latent space can be more effective than parameter updates, particularly for challenging out-of-distribution problems. This advantage stems from BLTO’s Bayesian formulation, which dynamically adapts exploration based on posterior uncertainty, enabling targeted reasoning adjustments that training-based methods cannot achieve at inference time.

4.3 Generalization to Other Domains

To verify generalization beyond mathematics, we evaluate BLTO on StrategyQA (Geva et al., 2021) for commonsense reasoning and Date Understanding (Srivastava et al., 2023) for symbolic reasoning. As shown in Table 3, BLTO achieves the best performance in three of the four settings and remains competitive on LLaMA-3.1-8B StrategyQA. On the symbolic reasoning task, BLTO attains 65.2% with LLaMA-3.1-8B, delivering a 9.6% improvement over Zero-Shot CoT, and reaches 76.8% with Qwen2.5-7B, outperforming all baselines. While training-based SoftCoT occasionally degrades performance on StrategyQA, BLTO consistently improves upon or matches the strongest

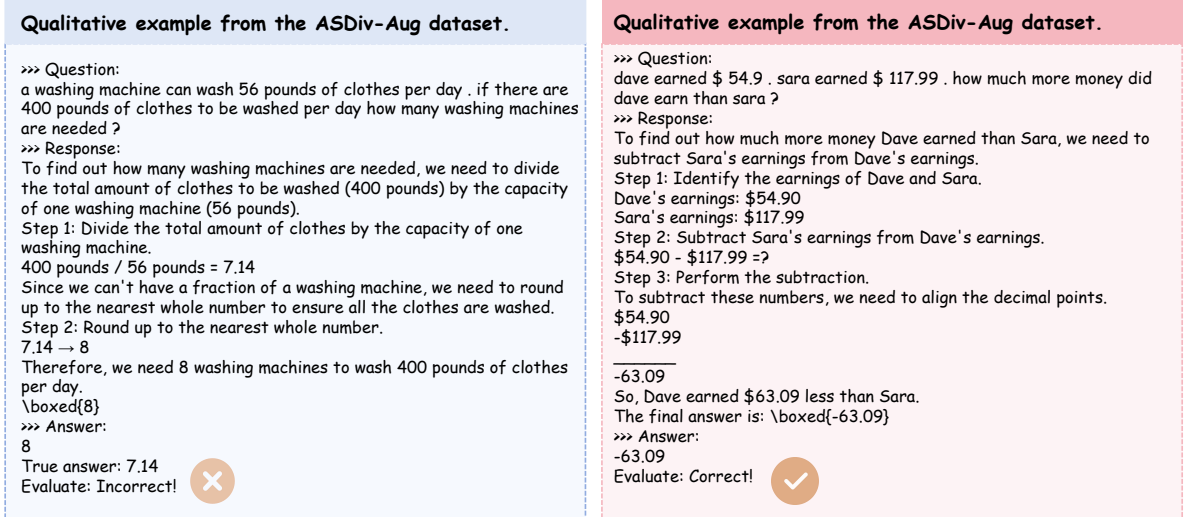


Figure 2: Qualitative examples from ASDiv-Aug illustrating BLTO’s reasoning capabilities.

Table 5: Ablation study on the number of latent thought tokens K . Accuracy (%) reported on ASDiv-Aug, GSM8K, and MATH-500 with LLaMA-3.1-8B-Instruct.

Tokens K	ASDiv-Aug	GSM8K	MATH-500
2	90.08	80.14	47.20
4	87.86	80.06	48.40
6	88.63	79.98	49.00
8	89.40	81.90	49.40
10	88.82	82.03	51.00

baselines. These results confirm that BLTO is a robust, domain-agnostic framework capable of enhancing reasoning across diverse task types.

4.4 Performance Across Different Model Sizes

Table 4 shows that our method consistently outperforms LTPO across Qwen2.5 model sizes on GSM8K and ASDiv-Aug. The improvement is most significant for smaller models (0.5B and 1.5B), with up to 25% higher accuracy, while gains for the 7B model are smaller. This demonstrates that our approach effectively enhances mathematical reasoning, especially for models with limited capacity.

4.5 Ablation Studies

Table 5 investigates the impact of latent thought token count K . Increasing K from 2 to 10 improves GSM8K performance, achieving 82.03% accuracy with $K = 10$. On ASDiv-Aug, the best performance of 90.08% occurs at $K = 2$ and remains stable thereafter, though intermediate values show slight degradation suggesting overparameterization for simpler problems. These

results indicate the optimal token count is task-dependent, with complex reasoning benefiting from greater representational capacity.

4.6 Qualitative Analysis

Figure 2 presents two representative examples from ASDiv-Aug. In the first, the question asks for the number of washing machines needed for 400 pounds of laundry given a 56-pound daily capacity. While $400 \div 56 = 7.14$ is mathematically accurate, fractional machines are impossible. BLTO correctly outputs 8, recognizing this real-world constraint, whereas the standard answer incorrectly expects 7.14. In the second, the model correctly computes $54.90 - 117.99 = -63.09$, accurately determining Dave earned less than Sara. These examples demonstrate BLTO’s ability to handle both pragmatic constraints and precise arithmetic.

5 Conclusion

We introduced Bayesian Latent Thought Optimization (BLTO), a probabilistic framework for test-time reasoning enhancement in frozen LLMs. By formulating latent thought optimization as Bayesian inference, BLTO maintains a variational distribution over reasoning states. The combination of entropy-based intrinsic reward and reparameterized gradient optimization enables low-variance updates and automatic exploration-exploitation trade-offs. Extensive experiments across mathematical, commonsense, and symbolic reasoning tasks demonstrate that BLTO consistently outperforms both training-based methods and prior approaches,

highlighting its potential to advance reliable and generalizable reasoning in large language models.

Limitations

Despite the promising results of Bayesian Latent Thought Optimization, two limitations warrant discussion. First, the reliance on uncertainty guided exploration introduces additional computational overhead during inference, as each reasoning step requires maintaining a variational distribution over latent states. This complexity may limit deployment in latency sensitive applications. Second, while BLTO shows strong results on models with 0.5 to 8 billion parameters, its scalability to larger foundation models remains to be validated.

Acknowledgements

We gratefully acknowledge the support of the National Key R&D Program of China (No. 2024YFC2707805), the National Natural Science Foundation of China (No. 62506050, No. 62176029), China Postdoctoral Science Foundation Funded Project (2024M763867), the Chongqing Science and Technology Innovation and Application Development projects (No. CSTB2023TIAD-KPX0064, No. CSTB2025TIAD-JM010), and the Open Competition Program of Chongqing Municipal Commission of Economy and Information Technology (No. YJX-202500100200X).

The experimental and computational work in this research runs on the Huawei Cloud AI Compute Service. We appreciate the stable compute supply from this platform.

References

- Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 225–237.
- Xinghao Chen, Anhao Zhao, Heming Xia, Xuan Lu, Hanlin Wang, Yanjun Chen, Wei Zhang, Jian Wang, Wenjie Li, and Xiaoyu Shen. 2025. Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. *arXiv preprint arXiv:2505.16782*.
- 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*.
- Yuntian Deng, Yejin Choi, and Stuart Shieber. 2024. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*.
- Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. Efficient reasoning models: A survey. *arXiv preprint arXiv:2504.10903*.
- Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, 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, 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, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.
- Jinu Lee and Julia Hockenmaier. 2025. Evaluating step-by-step reasoning traces: A survey. *arXiv preprint arXiv:2502.12289*.

- Hengli Li, Chenxi Li, Tong Wu, Xuekai Zhu, Yuxuan Wang, Zhaoxin Yu, Eric Hanchen Jiang, Song-Chun Zhu, Zixia Jia, Ying Nian Wu, and 1 others. 2025. Seek in the dark: Reasoning via test-time instance-level policy gradient in latent space. *arXiv preprint arXiv:2505.13308*.
- 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.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Avinash Patil and Aryan Jadon. 2025. Advancing reasoning in large language models: Promising methods and approaches. *arXiv preprint arXiv:2502.03671*.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, and 25 others. 2025. *Qwen2.5 technical report*. *Preprint*, arXiv:2412.15115.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.
- 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.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, and 1 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*.
- Guohao Sun, Hang Hua, Jian Wang, Jiebo Luo, So-hail Dianat, Majid Rabbani, Raghuveer Rao, and Zhiqiang Tao. 2025. Latent chain-of-thought for visual reasoning. *arXiv preprint arXiv:2510.23925*.
- Xinyi Wang, Nikunj Saunshi, Rui-Jie Zhu, Liu Yang, Yuntian Deng, Nishanth Dikkala, Jiaheng Liu, and Zhiyuan Li. Latent & implicit thinking—going beyond cot reasoning. In *ICLR 2026 Workshop Proposals*.
- 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.
- Fangzhi Xu, Hang Yan, Chang Ma, Haiteng Zhao, Qiushi Sun, Kanzhi Cheng, Junxian He, Jun Liu, and Zhiyong Wu. 2025a. Genius: A generalizable and purely unsupervised self-training framework for advanced reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13153–13167.
- 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.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- 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*.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.
- Weihaohao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Ke-qing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.