PARALLELIZABLE PROBABILISTIC LATENT REASONING FOR DIVERSE REASONING TRAJECTORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models relying on verbalized Chain-of-Thought (CoT) are constrained to token-by-token generation in the language space, where many tokens serve textual coherence rather than core reasoning. This entangles genuine differences in logical trajectories with superficial variations in verbalization. Moving reasoning into a compact, continuous latent space offers a promising alternative for efficiently representing and exploring diverse reasoning trajectories. However, current latent models remain largely deterministic, capturing only a single reasoning trajectory rather than modeling a distribution over multiple valid alternatives. While stochastic approaches parameterize distributions over latent reasoning steps, they frequently suffer from unstable optimization and limited empirical performance. Furthermore, their reliance on autoregressive latent prediction during training precludes effective parallelization, leading to prohibitively long training times. To overcome these bottlenecks, we propose *Probabilistic Latent Reasoning Models* (PLRM), a parallelizable variational framework for probabilistic latent reasoning. PLRM adopts a JEPA-style learning framework in which the posterior infers latent distributions in parallel from observed reasoning trajectories, while the prior learns to predict the subsequent latent distribution via teacher forcing. To model heterogeneous reasoning processes, we introduce a discrete strategy variable together with a mixture-based prior, allowing the model to preserve the Gaussian-mixture structure induced by multiple distinct reasoning trajectories rather than collapsing them into a single Gaussian. Across diverse benchmarks, PLRM achieves performance competitive with deterministic latent reasoning models on single-trajectory datasets while enabling efficient training and consistently outperforming competing stochastic baselines on multi-trajectory datasets. Codes are available here.

1 INTRODUCTION

Autoregressive language models have developed strong reasoning capabilities by learning to generate chain-of-thought (CoT) traces in natural language (Wei et al., 2022; Chung et al., 2024). This success has established language as the dominant interface for both supervising and executing intermediate reasoning. A key advantage is that next-token prediction exposes reasoning as an explicit generative trajectory, enabling the exploration of multiple hypotheses and solution paths for tasks such as mathematical proof search and scientific hypothesis generation (Wang et al., 2022; Yao et al., 2023; Trinh et al., 2024; Gottweis et al., 2026). It also enables reinforcement learning by optimizing generated reasoning trajectories from reward feedback (Ouyang et al., 2022).

However, despite its favorable training properties and strong empirical performance, language is not necessarily an efficient substrate for reasoning. Many tokens primarily serve linguistic coherence rather than core reasoning, yet current Large Language Models (LLMs) devote comparable computation to both reasoning-critical and purely linguistic tokens (Hao et al., 2024). This is not only introduces substantial computational overhead, but also entangles genuine differences in reasoning logic with superficial variations in how the same reasoning process is verbalized (Kuhn et al., 2023). Moreover, each reasoning step in language space is restricted to a discrete choice from a vocabulary of size v, limiting each sampled token to at most $\log_2(v)$ bits of information (Gozeten et al., 2026).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

These limitations raise a fundamental question: *in what representational space should reasoning unfold?* Latent reasoning models address this question by replacing language tokens with continuous representations. Coconut (Hao et al., 2024) uses curriculum learning to progressively replace CoT tokens with continuous latent representations, while CODI (Shen et al., 2025) and Sim-CoT (Wei et al., 2026) employ self-distillation and language reconstruction to stabilize the transition to latent reasoning. These approaches achieve competitive performance with CoT supervised fine-tuning (CoT-SFT) methods while requiring substantially fewer reasoning steps. However, this transition from language to latent space sacrifices several desirable properties of SFT. Most notably, existing latent reasoning models largely rely on deterministic latent transitions, reducing diverse reasoning paths to a single trajectory and thereby limiting their ability to model reasoning as a distribution over trajectories.

In this work, we introduce PLRM, a variational framework for probabilistic latent reasoning that moves reasoning from language to continuous latent space while preserving the key properties that make CoT-SFT effective. Rather than merely replacing tokens with latent representations, PLRM adopts a JEPA-style predictive objective: a posterior constructs target latent distributions from verbal CoT traces, while a problem-conditioned prior learns to predict them. By jointly optimizing both end-to-end, PLRM encourages the latent representations to retain core reasoning information while discarding incidental variation in verbalization. We further show that multiple valid reasoning trajectories for the same problem can induce a multimodal target distribution, and introduce a mixture-based prior to model this multimodality without collapsing it into a single Gaussian.

Experimentally, PLRM performs strongly across mathematical and commonsense reasoning tasks, substantially outperforming existing stochastic latent reasoning models while remaining competitive with deterministic latent reasoning methods under more efficient training. Performance is particularly strong on GSM8K-Aug-NL (Deng et al., 2023) and CommonsenseQA (Talmor et al., 2019), where many CoT tokens serve verbalization rather than core reasoning, consistently capturing core reasoning information in latent representations. To evaluate distributional modeling under diverse reasoning trajectories, we further construct the HiddenBranch dataset from ProofWriter (Tafjord et al., 2021), where each problem admits multiple reasoning paths and answers. On this dataset, PLRM substantially outperforms CoT-SFT and all stochastic latent reasoning baselines, demonstrating its ability to effectively model distributions over diverse reasoning trajectories.

2 Desiderata for Latent Reasoning

In this section, we first identify the desirable properties of conventional supervised fine-tuning, then examine how latent reasoning changes this paradigm, highlighting both the benefits it introduces and the properties it may sacrifice.

2.1 Properties from Conventional Supervised Fine-Tuning

Consider a language model that receives a problem x, generates a verbalized reasoning trace (CoT) $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_T)$, and then predicts the final answer y. Conventional supervised fine-tuning can be viewed as learning the following joint distribution:

$$p(y, \boldsymbol{\tau} \mid x) = p(y \mid \boldsymbol{\tau}, x) \cdot p(\boldsymbol{\tau} \mid x) \tag{1}$$

This formulation provides three desirable properties. **First**, next-token prediction is inherently stochastic, allowing the model to represent uncertainty and explore multiple reasoning trajectories through sampling. This breadth-wise exploration is essential for discovering diverse solutions (Wang et al., 2022) and also underlies the basis for reinforcement-learning-based optimization, where trajectories are sampled and optimized according to their rewards (Ouyang et al., 2022). **Second**, every observed reasoning token serves as a prediction target, providing dense, fine-grained supervision throughout the reasoning trajectory (Bengio et al., 2003). **Finally**, teacher forcing on the ground-truth trajectory enables parallel training, a property that has been fundamental to the efficient scaling of modern language models (Vaswani et al., 2017).

2.2 Shifting Reasoning from Language Space to Latent Space

To improve the inefficiency of language-space reasoning, latent reasoning models replace verbalized reasoning steps with a sequence of continuous latent variables (Hao et al., 2024). Concretely,

conventional language-space reasoning computes a vocabulary distribution from the final-layer hidden representation h_t and samples the next reasoning token: $h_t = \mathrm{Transformer}(x, \tau_{<t}), \quad p(\tau_t \mid x, \tau_{<t}) = \mathrm{Softmax}(W_{\mathrm{out}} h_t)$, where W_{out} denotes the output head of a language model. Latent reasoning instead maps the hidden representation directly to a continuous latent variable through a learned projection, $z_k = g_\psi(h_k)$, and uses the resulting latent variable as an intermediate reasoning step. Repeating this process provides a latent reasoning trajectory $\mathbf{z} = (z_1, \ldots, z_K)$ that replaces the verbalized reasoning trace $\boldsymbol{\tau}$, leading to the generic factorization $p(y, \mathbf{z} \mid x) = p(y \mid \mathbf{z}, x) p(\mathbf{z} \mid x)$.

Deterministic Latent Reasoning and Its Pitfalls. While latent reasoning removes intermediate verbalization, the three desirable properties above are no longer preserved by construction. A deterministic projection produces a single latent trajectory, foregoing probabilistic modeling of alternative reasoning paths and limiting the latent space's ability to represent diverse underlying reasoning processes. Moreover, the absence of target latent variables prevents dense supervision and teacher forcing, heavily bottlenecking the training process with sequential rollouts (Hao et al., 2024; Shen et al., 2025). SIM-CoT (Wei et al., 2026) partially addresses the supervision gap by reconstructing $\boldsymbol{\tau}$ from each latent step, but the underlying latent trajectories remain deterministic and still require slow, sequential execution during training.

Potential and Challenges of Stochastic Latent Reasoning. Unlike deterministic latent reasoning, stochastic modeling can better exploit the latent space to represent a distribution over alternative reasoning trajectories, distinguishing genuine differences in reasoning logic from superficial variations in verbalization. This enables breadth-wise exploration directly in the core reasoning space and provides a stochastic foundation for reinforcement learning over latent reasoning trajectories. A straightforward approach is to predict Gaussian parameters from the hidden representation and sample each latent variable as

$$(\mu_k, \sigma_k) = g_\psi(h_k), \qquad z_k \sim \mathcal{N}(\mu_k, \sigma_k^2). \tag{2}$$

However, introducing stochasticity alone does not determine what a meaningful latent reasoning distribution should represent. A useful target distribution should retain answer-relevant reasoning information while excluding incidental information that cannot be inferred from x. Existing methods typically impose guidance through heuristically constructed or externally derived targets: CoLaR (Tan et al., 2026) uses scaled sums of CoT token embeddings, while ReGuLaR (Wang et al., 2026) derives Gaussian priors from rendered CoT representations using a frozen visual encoder and a trainable adapter. Such targets are not explicitly learned to preserve answer-relevant core reasoning information while remaining predictable from x, and some approaches still require sequential latent rollout. Thus, the key challenge is not merely to make latent reasoning stochastic, but to learn a meaningful and predictable distribution over reasoning trajectories.

3 Probabilistic Latent Reasoning Models

Motivated by above, we introduce Probabilistic Latent Reasoning Models (PLRM), a variational framework that jointly learns target latent distributions and a problem-conditioned prior to predict them. We first develop a variational formulation that preserves the three desirable properties of conventional supervised fine-tuning. We then show that this formulation naturally takes the form of a JEPA-style learning framework, in which target latent distributions capture core reasoning information while abstracting away incidental variations in verbalization. Finally, we extend the framework to settings with multiple heterogeneous reasoning strategies by introducing an explicit strategy variable and a mixture-based prior capable of representing distinct modes of the reasoning distribution.

3.1 Variational Latent Reasoning

We define $\mathbf{z}$ as the latent reasoning trajectory underlying both the verbalized trace $\boldsymbol{\tau}$ and the final answer y, following the generative structure $x \to \mathbf{z} \to \{\boldsymbol{\tau}, y\}$. Under this formulation, the trace $\boldsymbol{\tau}$ is treated as one possible verbal realization from $\mathbf{z}$, *i.e.,* $\boldsymbol{\tau} \sim p_\theta(\boldsymbol{\tau} \mid \mathbf{z})$. Accordingly, the joint distribution in Eq. 1 can be reformulated by marginalizing over the latent reasoning trajectory as

$$p_\theta(y, \boldsymbol{\tau} \mid x) = \int p_\theta(\boldsymbol{\tau} \mid \mathbf{z})\, p_\theta(y \mid \mathbf{z}, x)\, p_\theta(\mathbf{z} \mid x)\, dz. \tag{3}$$

Here, $p_\theta(\mathbf{z} \mid x) = \prod_{k=1}^{K} p_\theta(z_k \mid x, z_{<k})$ represents the distribution over possible latent reasoning processes for a given problem. We retain x in $p_\theta(y \mid \mathbf{z}, x)$ for computational efficiency: since answer

generation follows the existing $(x, \mathbf{z})$ context, the model can continue decoding directly from the current KV cache without recomputing the context representations.

To benefit from fine-grained supervision and parallel training in the latent space, we introduce a factorized variational posterior $q_\phi(\mathbf{z} \mid x, \boldsymbol{\tau}) = \prod_{k=1}^{K} q_\phi(z_k \mid x, \boldsymbol{\tau})$. Because each posterior distribution is inferred directly from the observed reasoning trace, every latent step is associated with the target distribution without requiring sequential posterior inference. In our basic formulation, we parameterize both the posterior and prior at each latent step as Gaussian distributions:

$$q_\phi(z_k \mid x, \boldsymbol{\tau}) = \mathcal{N}\big(\mu_k^q, (\sigma_k^q)^2\big), \qquad p_\theta(z_k \mid x, z_{<k}) = \mathcal{N}\big(\mu_k^p, (\sigma_k^p)^2\big). \tag{4}$$

The framework is then jointly trained by optimizing the following Evidence Lower Bound (ELBO) of $\log p_\theta(y, \boldsymbol{\tau} \mid x)$:

$$\underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|x,\boldsymbol{\tau})}\left[\log p_\theta(\boldsymbol{\tau} \mid \mathbf{z})\right]}_{\text{CoT reconstruction}} + \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|x,\boldsymbol{\tau})}\left[\log p_\theta(y \mid \mathbf{z}, x)\right]}_{\text{answer prediction}} - \underbrace{D_{\mathrm{KL}}\left(q_\phi(\mathbf{z} \mid x, \boldsymbol{\tau}) \,\|\, p_\theta(\mathbf{z} \mid x)\right)}_{\text{distribution alignment}}. \tag{5}$$

Takeaway. Importantly, this variational formulation moves reasoning from language space to latent space while maintaining three desired properties of conventional supervised fine-tuning. **First**, both the posterior $q_\phi(\mathbf{z} \mid x, \boldsymbol{\tau})$ and the prior $p_\theta(\mathbf{z} \mid x)$ model distributions over latent trajectories, preserving stochasticity over alternative reasoning paths. **Second**, the posterior provides dense supervision by inferring a target distribution for every latent step directly from the observed verbal reasoning trace. **Finally**, since the KL objective is evaluated on posterior-generated latent trajectories, the prior performs teacher-forced next-latent distribution prediction, allowing all latent steps to be trained in parallel.

Having moved reasoning from language space to a continuous latent space, a natural question arises: *what makes a useful latent reasoning representation?* We argue that such a representation should satisfy two basic requirements. First, because reasoning ultimately serves to solve the problem, the latent trajectory should preserve sufficient information to predict the correct answer. Second, because the latent trajectory must be generated from the problem alone at inference time, the latent trajectory should remain predictable from the problem without access to the verbalized reasoning trace. Together, these requirements favor latent representations that capture core, problem-relevant reasoning information while abstracting away incidental details arising from a particular verbalization.

3.2 Joint Distribution Predictive Architecture

Our formulation in Eq. 5 captures these requirements through three complementary objectives:

① $p_\theta(\boldsymbol{\tau} \mid \mathbf{z})$: preserve information about the verbalized reasoning trace $\boldsymbol{\tau}$,
② $p_\theta(y \mid \mathbf{z}, x)$: support prediction of the final answer y, and
③ $D_{\mathrm{KL}}(q_\phi(\mathbf{z} \mid x, \boldsymbol{\tau}) \| \, p_\theta(\mathbf{z} \mid x))$: remain predictable from the prior conditioned on the problem x.

While the first two objectives ground the latent representations in task-relevant semantics, the third objective shapes the structure of the latent space itself. Specifically, the posterior acts as a target encoder that provides latent target distributions, while the prior acts as a predictor that learns to match these targets without access to the verbalized reasoning trace. This naturally gives rise to a JEPA-style framework, (Baevski et al., 2022; Assran et al., 2023), with the distinction that both the target and predicted representations are probability distributions rather than deterministic embeddings.

In the spirit of JEPA, which emphasizes predictive representations over reconstructing every detail of the input, our framework encourages the latent representation to capture core reasoning information while abstracting away incidental details of its verbal realization. The following proposition provides an information-theoretic interpretation of the role of the problem-conditioned prior $p_\theta(\mathbf{z} \mid x)$:

Proposition 1 (Problem-Conditioned Information Bottleneck) *Considering the expectation of distribution alignment over the data distribution $p(x, \boldsymbol{\tau})$, we obtain*

$$\begin{aligned}\mathbb{E}_{x,\boldsymbol{\tau}}\left[D_{\mathrm{KL}}(q_\phi(\mathbf{z} \mid x, \boldsymbol{\tau}) \,\|\, p_\theta(\mathbf{z} \mid x))\right] &= \mathbb{E}_{x,\boldsymbol{\tau},\mathbf{z}}\left[\log \frac{q_\phi(\mathbf{z} \mid x, \boldsymbol{\tau})}{\bar{q}_\phi(\mathbf{z} \mid x)}\right] + \mathbb{E}_{x,\mathbf{z}}\left[\log \frac{\bar{q}_\phi(\mathbf{z} \mid x)}{p_\theta(\mathbf{z} \mid x)}\right] \\ &= I_q(\mathbf{z}; \boldsymbol{\tau} \mid x) + \mathbb{E}_x\left[D_{\mathrm{KL}}(\bar{q}_\phi(\mathbf{z} \mid x) \,\|\, p_\theta(\mathbf{z} \mid x))\right], \end{aligned} \tag{6}$$

where $\bar{q}_\phi(\mathbf{z} \mid x) = \mathbb{E}_{\boldsymbol{\tau} \sim p(\boldsymbol{\tau} \mid x)}[q_\phi(\mathbf{z} \mid x, \boldsymbol{\tau})]$ denotes the aggregated posterior over latent reasoning trajectories for a given problem x. Thus, as the prior approaches the aggregated posterior, the alignment objective increasingly penalizes information about the particular verbalized trace that is not predictable from the problem.

Together with answer prediction loss $p_\theta(y \mid \mathbf{z}, x)$, this biases $\mathbf{z}$ toward retaining task-relevant reasoning information while suppressing trace-specific variation.

Preventing Representation Collapse. To prevent collapse of the target representations, recent JEPA methods employ representation regularization that shapes the embedding distribution toward a standard normal Gaussian (Balestriero & LeCun, 2025; Maes et al., 2026). Following this principle, we apply VISReg (Wu et al., 2026) to the posterior means μ^q to stabilize joint posterior–prior training. We regularize μ^q rather than sampled latent z, since Gaussian sampling noise could satisfy the regularization even when the posterior means collapse.

3.3 Modeling Heterogeneous Reasoning Strategies

As Proposition 1 shows, the prior is optimized to match the aggregated posterior $\bar{q}_\phi(\mathbf{z} \mid x)$. When a problem x has a single reasoning trace $\boldsymbol{\tau}$, $\bar{q}_\phi(\mathbf{z} \mid x)$ reduces to $q_\phi(\mathbf{z} \mid x, \boldsymbol{\tau})$, recovering the original distribution-alignment term. When x admits multiple reasoning trajectories with corresponding verbalized traces $\{\boldsymbol{\tau}_i\}_{i=1}^{N_x}$, marginalizing over all valid traces induces a multimodal target distribution. This multimodality can already arise at the first latent step: $\bar{q}_\phi(z_1 \mid x) = \mathbb{E}_{\boldsymbol{\tau} \sim p(\boldsymbol{\tau} \mid x)}[q_\phi(z_1 \mid x, \boldsymbol{\tau})] = \sum_{i=1}^{N_x} p(\boldsymbol{\tau}_i \mid x)\, \mathcal{N}(\mu_{i,1}^q, (\sigma_{i,1}^q)^2)$, whereas the basic prior is restricted to a single Gaussian, *i.e.,* $p_\theta(z_1 \mid x) = \mathcal{N}(\mu_1^p, (\sigma_1^p)^2)$.

This mismatch creates two complementary failure modes. First, the posterior can reduce the KL divergence by pulling trajectory-specific Gaussian modes toward a common region, collapsing distinct reasoning strategies in the latent space. Second, if the posterior preserves these distinct modes, the prior must cover them with a single Gaussian. Under the forward KL, this encourages the prior to assign probability mass across all modes, potentially inflating its variance and placing substantial mass between them. Thus, a unimodal prior can either *collapse the multimodal structure of the target representation* or *blur it in the predictive distribution*.

Strategy-Conditioned Mixture Modeling. To explicitly model multimodal reasoning, we introduce a categorical strategy variable $c \in [M]$ representing up to M distinct reasoning strategies. Rather than requiring $\mathbf{z}$ to represent all valid reasoning trajectories, c partitions heterogeneous trajectories into distinct strategy modes, within which $\mathbf{z}$ captures trajectory-level variation. We sample a strategy once as $c \sim p_\theta(c \mid x)$ and keep it fixed throughout the latent trajectory, providing the conceptual hierarchy $(x, c) \to \mathbf{z} \to (\boldsymbol{\tau}, y)$. Accordingly, we factorize the variational posterior to jointly infer the strategy and latent trajectory from the observed trace, *i.e.,* $q_\phi(c, \mathbf{z} \mid x, \boldsymbol{\tau}) = q_\phi(c \mid x, \boldsymbol{\tau}) \prod_{k=1}^{K} q_\phi(z_k \mid x, \boldsymbol{\tau})$. The resulting ELBO of $\log p_\theta(y, \boldsymbol{\tau} \mid x)$ is given by modifying only the distribution alignment term in Eq. 5 to incorporate the strategy c as

$$\mathbb{E}_{q_\phi(\mathbf{z} \mid x, \boldsymbol{\tau})}[\log p_\theta(\boldsymbol{\tau} \mid \mathbf{z})] + \mathbb{E}_{q_\phi(\mathbf{z} \mid x, \boldsymbol{\tau})}[\log p_\theta(y \mid \mathbf{z}, x)] - D_{\mathrm{KL}}(q_\phi(c, \mathbf{z} \mid x, \boldsymbol{\tau}) \,\|\, p_\theta(c, \mathbf{z} \mid x)) \,. \quad (7)$$

We use this ELBO as the training objective; details of its optimization is provided in Appendix A.1.

Following Proposition 1, averaging the KL term in Eq. 7 over the data distribution $p(x, \boldsymbol{\tau})$ yields the aggregated alignment term $\mathbb{E}_x[D_{\mathrm{KL}}(\bar{q}_\phi(c, \mathbf{z} \mid x) \,\|\, p_\theta(c, \mathbf{z} \mid x))]$. Focusing on a given problem x reveals why this parameterization naturally accommodates multimodal reasoning, as the point-wise KL divergence, $D_{\mathrm{KL}}(\bar{q}_\phi(c, \mathbf{z} \mid x) \,\|\, p_\theta(c, \mathbf{z} \mid x))$, decomposes into

$$D_{\mathrm{KL}}(\bar{q}_\phi(c \mid x) \,\|\, p_\theta(c \mid x)) + \mathbb{E}_{c \sim \bar{q}_\phi(c \mid x)}\left[D_{\mathrm{KL}}(\bar{q}_\phi(\mathbf{z} \mid x, c) \,\|\, p_\theta(\mathbf{z} \mid x, c))\right], \quad (8)$$

where $\bar{q}_\phi(\mathbf{z} \mid x, c)$ denotes the conditional distribution induced by the joint aggregated posterior. The strategy variable c separates heterogeneous reasoning modes, so each conditional target $\bar{q}_\phi(\mathbf{z} \mid x, c)$ only needs to capture variation within a single strategy. When c sufficiently separates the heterogeneous modes, each conditional target can be well approximated by a unimodal Gaussian, allowing the multimodal target distribution to be represented as a mixture of strategy-specific Gaussians.

3.4 Implementing with Autoregressive Language Models

PLRM builds on standard autoregressive language models and introduces lightweight modules for latent distribution prediction and latent injection, while leaving the backbone architecture unchanged. Implementation details are provided in Appendix A.

Shared Feature Transformation. At each latent step k, the language-model hidden representation h_k is mapped into an intermediate representation $r_k = \mathrm{SwiGLU}(\mathrm{RMSNorm}(h_k))$ (Shazeer, 2020; Zhang & Sennrich, 2019). The posterior and prior share the backbone and nonlinear feature transformation up to r_k, while using separate linear heads to parameterize their respective distributions.

Strategy-Conditioned Prior. For each strategy $m \in [M]$, the prioir constructs a strategy-specific representation using Feature-wise Linear Modulation (FiLM) (Perez et al., 2018), *i.e.,* $\tilde{r}_{m,k} = (1 + \gamma_m) \odot r_k + \beta_m$, where (γ_m, β_m) are learnable strategy-specific parameters shared across all latent steps. At the first latent step, a linear layer followed by a softmax predicts the strategy distribution $p_\theta(c \mid x)$. Two linear heads then map each $\tilde{r}_{m,k}$ to the mean and log-variance of a d-dimensional Gaussian, $(\mu^p_{m,k}, \log((\sigma^p_{m,k})^2))$, yielding

$$p_\theta(z_k \mid x, z_{<k}, c = m) = \mathcal{N}\left(\mu^p_{m,k}, (\sigma^p_{m,k})^2\right), \tag{9}$$

where d can be chosen smaller than the language-model hidden dimension to encourage a compact latent reasoning space. After sampling z_k, a linear projection followed by LayerNorm maps it back to the language-model hidden dimension. The resulting vector is then injected as the continuous input embedding for the subsequent latent step.

Parallel Target Latent Construction. To construct the target latent trajectory distribution in parallel, we append K copies of a Latent-Reasoning special token, `[LR]`, to $(x, \boldsymbol{\tau})$. A single forward pass *simultaneously* produces the hidden representation h_k at the k-th `[LR]` position, which is transformed into r_k using the shared feature transformation and then mapped by two linear heads to parameterize $q_\phi(z_k \mid x, \boldsymbol{\tau}) = \mathcal{N}(\mu^q_k, (\sigma^q_k)^2)$.

4 Related Works

Having discussed autoregressive language models, including conventional supervised fine-tuning and latent reasoning formulations, in Section 2, we review here several additional lines of work that inform our approach.

Joint-Embedding Predictive Architecture. JEPA learns latent representations by predicting target embeddings rather than reconstructing every input detail, encouraging high-level semantic information (Baevski et al., 2022; Assran et al., 2023). This predictive latent space has subsequently been exploited for planning and reasoning (Maes et al., 2026; Liu et al., 2025a). However, these approaches operate on deterministic latent embeddings rather than explicitly modeling distributions over alternative trajectories.

In particular, JEPA-Reasoner (Liu et al., 2025a) first learns a deterministic latent reasoning trajectory and then freezes the Reasoner while separately training a Talker to reconstruct the corresponding reasoning and answer sequence. Consequently, inference requires both autoregressive latent rollout and a subsequent verbalization stage, rather than predicting the answer directly from the latent trajectory. In contrast, PLRM learns a probabilistic latent trajectory distribution in an end-to-end manner and predicts the final answer directly from the latent trajectory without requiring CoT generation.

Alternative Generative Paradigms for Latent Reasoning. A separate line of work redesigns the latent reasoning process itself: LaDiR (Kang et al., 2026) performs reasoning through latent diffusion, while MarCos (Liu et al., 2025b) models reasoning as a hidden Markov chain of continuous thoughts. These methods first impose a latent space through auxiliary objectives on verbalized reasoning, such as VAE reconstruction with a Gaussian prior or sparsity regularization, and subsequently learn the corresponding latent reasoning dynamics. As a result, what information the latent representations retain is largely shaped by these predefined objectives rather than being jointly optimized for answer prediction and predictability from the problem. In contrast, PLRM jointly learns the target distributions and the problem-conditioned prior, allowing the latent reasoning distribution itself to adapt throughout training.

Table 1: Accuracy (%) and token usage of GPT-2 across in-distribution (ID) and out-of-distribution (OOD) tasks.

Type	Method	ID Tasks						OOD Tasks			
		GSM8K-NL		GSM8K		CSQA		Accuracy (%)			Avg.
		Acc.	# Tokens	Acc.	# Tokens	Acc.	# Tokens	SVAMP	GSM-Hard	Multi-Arith	# Tokens (OOD)
Verbalized	CoT-SFT	30.93	67.00	39.88	31.50	34.81	67.60	34.20	6.70	81.80	42.60
Deterministic	COCONUT	21.90	40.80	29.64	26.70	27.40	57.90	24.10	4.40	30.83	28.60
	CODI	35.03	11.30	**41.17**	11.20	38.98	11.00	34.50	8.20	83.00	11.60
	SIM-CoT	17.20	11.20	38.97	11.30	33.58	11.00	22.70	4.02	26.50	11.70
Stochastic	ReGuLaR	19.32	6.30	17.89	6.40	20.10	12.40	22.38	3.79	31.87	5.70
	CoLaR	20.32	33.20	36.39	17.30	21.62	16.50	25.70	4.09	54.17	24.60
	PLRM (Ours)	**36.30**	11.28	38.40	11.25	**43.40**	11.00	**34.60**	**8.26**	**88.83**	11.63

Table 2: Accuracy (%) and token usage of Llama-3.2-1B-Instruct across in-distribution (ID) and out-of-distribution (OOD) tasks.

Type	Method	ID Tasks						OOD Tasks			
		GSM8K-NL		GSM8K		CSQA		Accuracy (%)			Avg.
		Acc.	# Tokens	Acc.	# Tokens	Acc.	# Tokens	SVAMP	GSM-Hard	Multi-Arith	# Tokens (OOD)
Verbalized	CoT-SFT	**53.22**	74.05	**58.91**	29.67	63.39	75.00	**58.60**	9.33	94.17	46.98
Deterministic	COCONUT	33.13	45.50	48.93	27.58	70.84	10.35	50.60	7.96	83.83	33.41
	CODI	42.76	13.21	54.44	13.19	67.32	12.00	56.00	9.40	**94.83**	13.49
	SIM-CoT	31.69	13.23	55.19	13.19	70.68	12.00	45.50	7.20	65.00	13.48
Stochastic	ReGuLaR	18.60	7.46	35.38	7.67	59.60	8.14	40.76	3.91	35.43	6.95
	CoLaR	29.34	37.07	44.35	17.17	42.51	23.63	39.40	6.75	75.33	28.20
	PLRM (Ours)	42.08	13.29	50.00	13.23	**74.28**	12.00	49.40	**10.01**	**94.83**	13.62

5 EXPERIMENTS

We conduct experiments on mathematical and commonsense reasoning tasks using GPT-2 (Radford et al., 2019) and Llama-3.2-1B-Instruct (Grattafiori et al., 2024), evaluating PLRM in both in-domain and out-of-domain settings. We further consider a multi-trajectory reasoning setting in which each problem admits multiple valid reasoning trajectories with corresponding answers.

5.1 EXPERIMENTAL SETUP

Datasets. Following previous work (Shen et al., 2025), we train PLRM on three in-domain (ID) benchmarks: GSM8K-Aug (Deng et al., 2023), its natural-language variant GSM8K-Aug-NL, and CommonsenseQA (Talmor et al., 2019). For models trained on GSM8K-Aug-NL, we evaluate out-of-domain (OOD) generalization on three mathematical reasoning benchmarks: SVAMP (Patel et al., 2021), GSM-Hard (Gao et al., 2023), and MultiArith (Roy & Roth, 2015).

HiddenBranch Dataset. To evaluate whether PLRM can capture heterogeneous valid reasoning trajectories, we construct HiddenBranch from the open-world abductive reasoning setting of ProofWriter (Tafjord et al., 2021). HiddenBranch asks the model to find a hypothesis that makes a target conclusion provable from the given facts and rules. For an input x, there may exist multiple valid reasoning trajectories with corresponding answers, $\{(\boldsymbol{\tau}_i, y_i)\}_{i=1}^{N_x}$, where each $\boldsymbol{\tau}_i$ includes a different valid hypothesis and the reasoning process induced by that hypothesis. Thus, the same input can admit multiple heterogeneous reasoning trajectories and answers.

Baselines. We use verbalized CoT-SFT as a standard language-space reasoning baseline and compare PLRM with recent latent reasoning methods, which we group into deterministic and stochastic approaches. The deterministic baselines include COCONUT (Hao et al., 2024), CODI (Shen et al., 2025), and SIM-CoT (Wei et al., 2026), while the stochastic baselines include ReGuLaR (Wang et al., 2026) and CoLaR (Tan et al., 2026).

Further details on the datasets, the construction of *HiddenBranch*, implementation details, and evaluation protocols are provided in Appendix B.

5.2 Main Results

Equation-Dominant vs Verbalized Reasoning. We distinguish two single-trajectory reasoning regimes according to how reasoning is expressed in the supervision. GSM8K-Aug primarily represents intermediate reasoning through mathematical expressions, such that most tokens directly encode the underlying computation. In contrast, the verbalized reasoning traces used for GSM8K-Aug-NL and CommonsenseQA contain richer natural-language explanations, where many tokens serve verbalization in addition to core reasoning.

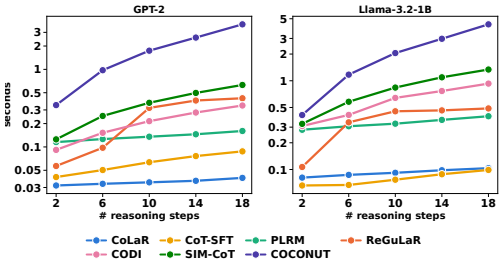

Figure 1: Average training time per iteration as the number of reasoning steps increases. For CoT-SFT, each step corresponds to four tokens, reflecting its approximately $4\times$ longer reasoning sequences.

This distinction is particularly important for latent reasoning. When the supervision is dominated by mathematical expressions, transferring the reasoning process into latent space mainly requires preserving the same computational content. In contrast, for verbalized reasoning traces, the model must compress a much longer sequence of T tokens into only K latent steps ($K \ll T$), requiring it to retain core reasoning information while abstracting away trace-specific variation. As shown in Tables 1 and 2, PLRM remains competitive with deterministic latent reasoning methods on GSM8K-Aug, while showing particularly strong performance on GSM8K-Aug-NL and CommonsenseQA. Being trained on GSM8K-Aug-NL, PLRM also generalizes strongly to OOD mathematical reasoning tasks, achieving the best or tied-best accuracy in five of six model–task settings. This strong OOD generalization suggests that PLRM captures transferable reasoning rather than overfitting to specific patterns of verbalized CoT Shen et al. (2025).

Training Efficiency. Figure 1 compares how the training cost scales with the number of reasoning steps. Methods trained with teacher forcing, including CoT-SFT, CoLaR, and PLRM, can process intermediate reasoning steps in parallel, resulting in only modest increases in training time as the number of latent steps grows. In contrast, all deterministic latent reasoning baselines (COCONUT, CODI, and SIM-CoT), together with the stochastic baseline (ReGuLaR), require sequential latent rollout during training, causing training cost to grow linearly with the number of latent steps. CoLaR is even faster because its target latent representations are constructed directly from summed token embeddings without an additional forward pass, but this efficiency comes at the cost of relying on heuristic targets and substantially lower empirical performance.

Importance of Joint Posterior–Prior Training. We further examine whether jointly learning the posterior and prior is necessary. Following a two-stage VAE setup Kang et al. (2026), we first train the posterior on CommonsenseQA using GPT-2 backbone to reconstruct the verbalized reasoning trace while fixing the prior to a standard normal Gaussian. We then extract latent trajectories for the entire training set, freeze the posterior, and train the problem-conditioned prior to predict these fixed targets. We additionally consider a variant that includes answer prediction when learning the posterior, encouraging the latent representations to retain answer-relevant information. As shown in Table 3, the two-stage variants achieve only 18.35–19.49% accuracy, comparable to CoLaR (21.62%), suggesting a common limitation of learning from fixed latent targets. In contrast, PLRM achieves 43.40%, highlighting the importance of jointly adapting the target latent distribution and its problem-conditioned predictor.

Table 3: Ablation study of PLRM.

Methods	Acc. (%)
Fixed Posterior	18.35
+ Answer Loss	19.49
CoLaR	21.62
PLRM (Ours)	**43.40**

5.3 Heterogeneous Reasoning Strategies

We next examine whether PLRM can model heterogeneous reasoning strategies when the same input admits multiple valid reasoning trajectories, as discussed in Section 3.3. We assess how well

the learned distribution captures these alternative hypotheses using **Hit Rate** and **Avg. TV Score**. Hit Rate measures whether greedy decoding recovers any valid answer. Avg. TV Score measures the agreement between the distribution of answers obtained from diverse reasoning trajectories and the distribution over all valid answers, jointly reflecting invalid generations and incomplete coverage of valid hypotheses; details are provided in Appendix B.2.

Visualizing Multimodal Latent Distributions. We first compare the single-component prior in Eq. 5 ($M = 1$) with our mixture-based prior using two components ($M = 2$). As shown in Figure 2, the Mono Prior initially expands its variance to cover distinct posterior modes. As reasoning proceeds, however, the posterior modes are progressively pulled together, collapsing the heterogeneous trajectories into a similar latent distribution. In contrast, the Double Prior assigns separate prior components to different posterior modes and maintains their separation across latent steps, preserving the multimodal structure without forcing heterogeneous reasoning trajectories to collapse. Additional visualizations for $M \geq 3$ and qualitative examples reconstructing latent reasoning trajectories into verbalized reasoning traces are provided in Appendix C.2.

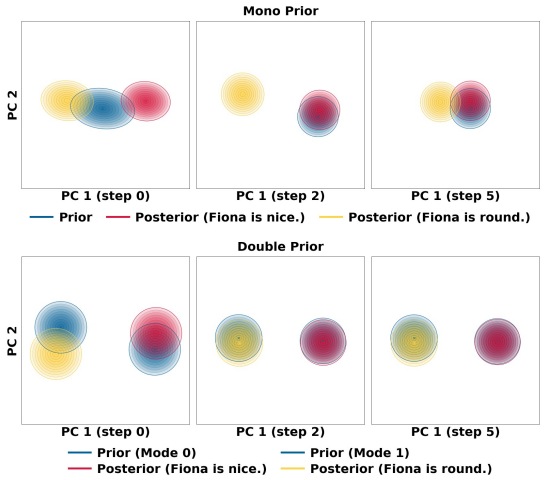

Figure 2: PCA visualization of prior and posterior latent distributions under mono (M=1) and double-prior (M=2) settings.

Quantitative Evaluation. We compare PLRM with CoT-SFT and stochastic latent reasoning baselines, which can represent distributions over reasoning trajectories. As shown in Table 4, it achieves the best performance on both Hit Rate and Avg. TV Score, improving the latter from 0.6160 to 0.7097 over CoT-SFT, a 15.2% relative gain that indicates better coverage of heterogeneous valid hypotheses while maintaining high deterministic accuracy. Table 5 examines the number of mixture components. While a single mode achieves high Hit Rate, its substantially lower TV Score indicates limited coverage of alternative valid hypotheses. Using two modes yields the best TV Score, indicating that the heterogeneity in HiddenBranch can be sufficiently modeled with two strategy components.

6 Conclusion

In this work, we proposed PLRM, a variational framework for probabilistic latent reasoning with dense supervision and parallel teacher-forced training. By jointly learning posterior target distributions and a problem-conditioned prior, PLRM captures core reasoning information while abstracting away trace-specific verbalization, and models heterogeneous strategies through a mixture-based prior. Across mathematical and commonsense reasoning tasks, PLRM outperforms existing stochastic latent reasoning methods while remaining competitive with deterministic approaches. We envision PLRM as a step toward scalable probabilistic reasoning in continuous latent space beyond verbalized chain-of-thought.

Table 4: Hit rate and average TV Score across different methods.

Metric	CoT-SFT	ReGuLaR	CoLaR	PLRM
Hit Rate	0.9116	0.8665	0.7397	**0.9205**
Avg. TV Score	0.6160	0.4753	0.3638	**0.7097**

Table 5: Ablation on the number of M.

Metric	M=1	M=2	M=3	M=4
Hit Rate	**0.9370**	0.9205	0.9292	0.9300
Avg. TV Score	0.6252	**0.7097**	0.6963	0.6908

AI USE STATEMENT

In this work, we used generative Al tools to aid and polish writing. We did not use generative AI tools for any other tasks requiring disclosure under the ICLR policy. We reviewed all AI-assisted content and take full responsibility for the final content of this work.

ETHICS STATEMENT

This work uses publicly available datasets and involves no direct interaction with human subjects. We additionally construct HiddenBranch by transforming examples from the existing ProofWriter dataset to create multiple valid reasoning trajectories for the same input. We plan to publicly release the resulting dataset and its construction procedure to support reproducibility and future research.

REPRODUCIBILITY STATEMENT

We will publicly release all code and datasets used in this work to facilitate reproducibility.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International conference on machine learning*, pp. 1298–1312. PMLR, 2022.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of machine learning research*, 6(Oct):1705–1749, 2005.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of machine learning research*, 25(70):1–53, 2024.

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.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International conference on machine learning*, pp. 10764–10799. PMLR, 2023.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, et al. Accelerating scientific discovery with co-scientist. *Nature*, pp. 1–3, 2026.

Alperen Gozeten, Muhammed Ildiz, Xuechen Zhang, Hrayr Harutyunyan, Ankit Singh Rawat, and Samet Oymak. Continuous chain of thought enables parallel exploration and reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 82881–82914, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

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.

Haoqiang Kang, Yizhe Zhang, Nikki Kuang, Nicklas Majamaki, Navdeep Jaitly, Yian Ma, and Lianhui Qin. Ladir: Latent diffusion enhances llms for text reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 81705–81733, 2026.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

Bingyang Kelvin Liu, Ziyu Patrick Chen, and David P Woodruff. Jepa-reasoner: Decoupling latent reasoning from token generation. *arXiv preprint arXiv:2512.19171*, 2025a.

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

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 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*, pp. 2080–2094, 2021.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. Codi: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693, 2025.

Wenxian Shi, Hao Zhou, Ning Miao, and Lei Li. Dispersed exponential family mixture vaes for interpretable text generation. In *International Conference on Machine Learning*, pp. 8840–8851. PMLR, 2020.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the association for computational linguistics: ACL-IJCNLP 2021*, pp. 3621–3634, 2021.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019.

Wenhui Tan, Jiaze Li, Jianzhong Ju, Zhenbo Luo, Ruihua Song, and Jian Luan. Think silently, think fast: Dynamic latent compression of llm reasoning chains. *Advances in Neural Information Processing Systems*, 38:4646–4668, 2026.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

Naonori Ueda and Ryohei Nakano. Deterministic annealing em algorithm. *Neural networks*, 11(2): 271–282, 1998.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Fanmeng Wang, Haotian Liu, Guojiang Zhao, Hongteng Xu, and Zhifeng Gao. Regular: Variational latent reasoning guided by rendered chain-of-thought. *arXiv preprint arXiv:2601.23184*, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

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.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. Sim-cot: Supervised implicit chain-of-thought. In *International Conference on Learning Representations*, volume 2026, pp. 56721–56742, 2026.

Haiyu Wu, Randall Balestriero, and Morgan Levine. Visreg: Variance-invariance-sketching regularization for jepa training. *arXiv preprint arXiv:2606.02572*, 2026.

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

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

A Implementation Details

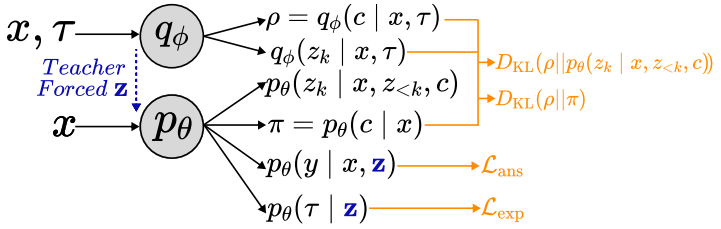

Figure 3: Overall Training Computation

A.1 Empirical Optimization

The objective in Eq. equation 7 contains the following trajectory-level KL divergence:

$$D_{\mathrm{KL}}(q_\phi(c, \mathbf{z} \mid x, \boldsymbol{\tau}) \,\|\, p_\theta(c, \mathbf{z} \mid x)) \,. \tag{10}$$

Using the factorization of the posterior and prior, this term can be decomposed as

$$\begin{aligned} &D_{\mathrm{KL}}(q_\phi(c, \mathbf{z} \mid x, \boldsymbol{\tau}) \,\|\, p_\theta(c, \mathbf{z} \mid x)) \\ &= D_{\mathrm{KL}}(q_\phi(c \mid x, \boldsymbol{\tau}) \,\|\, p_\theta(c \mid x)) \\ &\quad + \mathbb{E}_{q_\phi(c|x,\boldsymbol{\tau})} \sum_{k=1}^{K} \mathbb{E}_{q_\phi(z_{<k}|x,\boldsymbol{\tau})} \left[D_{\mathrm{KL}}(q_\phi(z_k \mid x, \boldsymbol{\tau}) \,\|\, p_\theta(z_k \mid x, z_{<k}, c)) \right] . \end{aligned} \tag{11}$$

The expectation over $z_{<k}$ is estimated by Monte Carlo samples from the posterior. We therefore focus on the inner KL term for a fixed strategy $c = m$. Both the posterior and prior are parameterized as diagonal Gaussian distributions:

$$q_\phi(z_k \mid x, \boldsymbol{\tau}) = \mathcal{N}\big(\mu_q^k, \mathrm{diag}\big((\sigma_q^k)^2\big)\big) \,, \tag{12}$$

$$p_\theta(z_k \mid x, z_{<k}, c = m) = \mathcal{N}\big(\mu_{p,m}^k, \mathrm{diag}\big((\sigma_{p,m}^k)^2\big)\big) \,. \tag{13}$$

For two d-dimensional Gaussian distributions, $q = \mathcal{N}(\mu_q, \Sigma_q)$ and $p = \mathcal{N}(\mu_p, \Sigma_p)$, the KL divergence is

$$D_{\mathrm{KL}}(q \| p) = \frac{1}{2} \left[\log \frac{|\Sigma_p|}{|\Sigma_q|} - d + \mathrm{tr}\big(\Sigma_p^{-1} \Sigma_q\big) + (\mu_p - \mu_q)^\top \Sigma_p^{-1} (\mu_p - \mu_q) \right] . \tag{14}$$

Since both covariance matrices are diagonal,

$$\log \frac{|\Sigma_p|}{|\Sigma_q|} = \sum_{j=1}^{d} \log \frac{(\sigma_{p,m,j}^k)^2}{(\sigma_{q,j}^k)^2} , \tag{15}$$

$$\mathrm{tr}\big(\Sigma_p^{-1} \Sigma_q\big) = \sum_{j=1}^{d} \frac{(\sigma_{q,j}^k)^2}{(\sigma_{p,m,j}^k)^2} , \tag{16}$$

$$(\mu_p - \mu_q)^\top \Sigma_p^{-1} (\mu_p - \mu_q) = \sum_{j=1}^{d} \frac{(\mu_{q,j}^k - \mu_{p,m,j}^k)^2}{(\sigma_{p,m,j}^k)^2} . \tag{17}$$

Substituting these terms gives the closed-form Gaussian KL at latent step k for strategy m:

$$D_{\mathrm{KL}}(q_\phi \,\|\, p_\theta) = \frac{1}{2} \sum_{j=1}^{d} \left[\log \frac{(\sigma_{p,m,j}^k)^2}{(\sigma_{q,j}^k)^2} + \frac{(\sigma_{q,j}^k)^2 + (\mu_{q,j}^k - \mu_{p,m,j}^k)^2}{(\sigma_{p,m,j}^k)^2} - 1 \right] . \tag{18}$$

Using this closed form, the expectation over the discrete strategy variable in Eq. equation 11 can be written as a responsibility-weighted sum:

$$\mathbb{E}_{q_\phi(c|x,\boldsymbol{\tau})} \sum_{k=1}^{K} \mathbb{E}_{q_\phi(z_{<k}|x,\boldsymbol{\tau})} \left[D_{\mathrm{KL}}(q_\phi(z_k \mid x, \boldsymbol{\tau}) \,\|\, p_\theta(z_k \mid x, z_{<k}, c)) \right]$$
$$= \sum_{k=1}^{K} \mathbb{E}_{q_\phi(z_{<k}|x,\boldsymbol{\tau})} \left[\sum_{m=1}^{M} \rho_m D_{\mathrm{KL}}(q_\phi(z_k \mid x, \boldsymbol{\tau}) \,\|\, p_\theta(z_k \mid x, z_{<k}, c = m)) \right], \tag{19}$$

where $\rho_m = q_\phi(c = m \mid x, \boldsymbol{\tau})$. The remaining expectation over $z_{<k}$ is approximated using posterior Monte Carlo samples.

A.2 Practical Details of PLRM

Although the formulation in Section 3 specifies the main training objectives of PLRM, we employ several additional regularization and optimization techniques to stabilize the joint training of the posterior and mixture prior. For brevity, throughout this subsection we denote

$$q_k \equiv q_\phi(z_k \mid x, \boldsymbol{\tau}), \qquad p_{k,m} \equiv p_\theta(z_k \mid x, z_{<k}, c = m),$$

and let $\rho_m \equiv q_\phi(c = m \mid x, \boldsymbol{\tau})$ denote the EM-style responsibility defined from the first latent step.

EM-style Strategy Learning. Inspired by divergence-based soft clustering (Banerjee et al., 2005), we use an EM-style soft assignment to stabilize strategy learning. Rather than introducing a separate classifier for c, we perform an E-step:

$$\rho_m \propto \exp\left(-\frac{1}{T} D_{\mathrm{KL}}(q_\phi(z_1 \mid x, \boldsymbol{\tau}) \,\|\, p_\theta(z_1 \mid x, c = m)) \right), \tag{20}$$

where T controls the assignment sharpness (Ueda & Nakano, 1998). The resulting responsibilities are fixed across the latent trajectory and used to update the corresponding strategy-specific prior components and $p_\theta(c \mid x)$, analogous to a gradient-based M-step. We infer the strategy from the first latent distribution because it is the only prior prediction made before any teacher-forced posterior latent is available. This restriction does not limit the intended multimodal modeling, since distinct reasoning strategies can already induce different modes at z_1, as shown above.

Mixture-Preserving KL Regularization. A direct responsibility-weighted Gaussian alignment, $\sum_m \rho_m D_{\mathrm{KL}}(q_k \| p_{k,m})$, encourages multiple prior components assigned to the same posterior q_k to move toward that common posterior target, thereby reducing the separation between mixture components. To isolate this contraction effect, we adopt the exponential-family decomposition derived in DEM-VAE (Shi et al., 2020):

$$\sum_m \rho_m D_{\mathrm{KL}}(q_k \| p_{k,m}) = D_{\mathrm{KL}}(q_k \| p_{\bar{\eta}_k}) + L_{d,k}, \tag{21}$$

where $\bar{\eta}_k = \sum_m \rho_m \eta_{m,k}$ is the responsibility-weighted average of the natural parameters of $p_{k,m}$, and $p_{\bar{\eta}_k}$ denotes the corresponding Gaussian distribution. The dispersion term $L_{d,k}$ is given by

$$L_{d,k} = \sum_m \rho_m A(\eta_{m,k}) - A(\bar{\eta}_k) \geq 0, \tag{22}$$

where $A(\cdot)$ denotes the Gaussian log-partition function. The non-negative term $L_{d,k}$ captures the additional cost induced by dispersion among the responsibility-weighted prior components and vanishes when these components coincide. Consequently, minimizing the original responsibility-weighted KL also penalizes separation among the prior components, introducing pressure toward component collapse. To attenuate this effect, we optimize the corrected Gaussian alignment

$$\mathcal{L}_{\mathrm{GKL},k} = \sum_m \rho_m D_{\mathrm{KL}}(q_k \| p_{k,m}) - \gamma L_{d,k}, \tag{23}$$

which reduces the component-collapse pressure induced by responsibility-weighted KL alignment while retaining a non-negative distribution-matching objective for $\gamma \in [0, 1]$.

Following the KL-balancing strategy of DreamerV2 (Hafner et al., 2020), we further stabilize training by separately controlling the gradients flowing to the predictive prior and target posterior. For each component-wise Gaussian KL term, we use

$$0.8\, D_{\mathrm{KL}}(\mathrm{sg}(q_k) \,\|\, p_{k,m}) + 0.2\, D_{\mathrm{KL}}(q_k \,\|\, \mathrm{sg}(p_{k,m}))\,, \tag{24}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. This places greater optimization pressure on the predictive prior to follow the posterior target.

Posterior Representation Regularization. As described in the main text, we apply VISReg to the posterior means rather than sampled latent variables. Concretely, VISReg consists of three complementary terms. The *center* term keeps the batch-aggregated posterior means close to zero, the *scale* term encourages each latent dimension to maintain unit batch-wise standard deviation, and the *shape* term encourages the standardized posterior means to follow a standard Gaussian under random one-dimensional projections. The resulting regularizer is

$$\mathcal{L}_{\mathrm{VISReg}} = \mathcal{L}_{\mathrm{scale}} + \mathcal{L}_{\mathrm{shape}} + \mathcal{L}_{\mathrm{center}}. \tag{25}$$

For the scale term, we use a squared deviation from unit standard deviation, $(1-\sigma)^2$, rather than a hinge penalty, so that the regularizer continues to provide a gradient when the batch standard deviation exceeds the target value. For the shape term, standardized posterior means are projected onto random unit directions, sorted across the batch, and matched to the corresponding quantiles of a standard Gaussian.

Overall Training Objective. With the dispersion correction, the practical distribution-alignment term is

$$\mathcal{L}_{\mathrm{align}} = D_{\mathrm{KL}}(\rho \,\|\, p_\theta(c \mid x)) + \frac{1}{K}\sum_{k=1}^{K}\left[\sum_m \rho_m D_{\mathrm{KL}}(q_k \| p_{k,m}) - \gamma L_{d,k}\right], \tag{26}$$

where the categorical KL is applied once per trajectory, whereas the Gaussian alignment is averaged over the K latent steps. The resulting training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{ans}} + \mathcal{L}_{\mathrm{exp}} + \beta(1-\lambda_{\mathrm{reg}})\mathcal{L}_{\mathrm{align}} + \beta\lambda_{\mathrm{reg}}\mathcal{L}_{\mathrm{VISReg}}. \tag{27}$$

We linearly warm up the shared coefficient β during training. Detailed values of the loss coefficients and training hyperparameters are provided in the following subsection.

A.3 Method Configuration

Table 6 summarizes the main optimization and method-specific hyperparameters used for PLRM. Unless otherwise specified, we use $K = 6$ latent reasoning steps and optimize all models with AdamW using cosine learning-rate decay and BF16 mixed precision. For the main benchmarks, including GSM8K, GSM8K-NL, and CommonsenseQA, we use a single prior component ($M = 1$), as each example provides a single reasoning trajectory. Dataset- and backbone-specific settings, including the learning rate, number of epochs, batch size, posterior context allocation, and objective coefficients, are reported in the table.

Notably, we use the same KL- and VISReg-related hyperparameters across different datasets and backbone architectures without per-setting tuning. Their consistent effectiveness across these settings suggests that PLRM is not highly sensitive to these regularization choices. In contrast, the posterior-token allocation per latent step (PTL) is adjusted by dataset to account for differences in reasoning-trace length, such that later latent targets are conditioned on progressively larger portions of the CoT trajectory. We fix the Gaussian latent dimension to $d = 768$ across all datasets and backbone architectures.

Table 6: Main training and method-specific hyperparameters of PLRM. PTL denotes the number of posterior-visible CoT tokens per latent step.

	GPT-2			Llama-3.2-1B		
Hyperparameter	**GSM8K**	**GSM8K-NL**	**CQA**	**GSM8K**	**GSM8K-NL**	**CQA**
Optimization						
Learning rate	1×10^{-3}	1×10^{-3}	2×10^{-3}	3×10^{-4}	3×10^{-4}	3×10^{-4}
Epochs	40	40	40	40	40	10
Effective batch size	128	128	128	256	256	256
Latent Configuration						
Posterior tokens per latent (PTL)	8	12	24	7	14	24
Max. input tokens	200	256	1000	200	256	1000
Objective and Regularization						
KL coefficient $\beta_{\max}$	3.0	3.0	3.0	3.0	3.0	3.0
KL warmup ratio	0.8	0.8	0.8	0.8	0.8	0.8
KL balance α	0.8	0.8	0.8	0.8	0.8	0.8
VISReg weight λ_{reg}	0.9	0.9	0.9	0.9	0.9	0.9
VISReg projections	512	512	512	512	512	512

B EXPERIMENTAL DETAILS

B.1 BENCHMARK DATASETS

In-Distribution Datasets. Following prior latent reasoning work (Shen et al., 2025), we consider three in-distribution (ID) benchmarks spanning mathematical and commonsense reasoning.

GSM8K-Aug. GSM8K-Aug (Deng et al., 2023) is an augmented version of the GSM8K mathematical reasoning benchmark. It consists of grade-school-level arithmetic word problems paired with intermediate reasoning traces and final answers. We use this dataset to evaluate latent reasoning in a standard mathematical reasoning setting.

GSM8K-Aug-NL. GSM8K-Aug-NL is the natural-language variant of GSM8K-Aug introduced in the same experimental setting (Deng et al., 2023). Compared with the standard GSM8K-Aug formulation, it provides reasoning trajectories in a more natural-language-oriented form. We use GSM8K-Aug-NL as an additional mathematical reasoning benchmark and use models trained on this dataset for our out-of-distribution evaluation.

CommonsenseQA. CommonsenseQA (CSQA) (Talmor et al., 2019) is a multiple-choice question answering benchmark designed to evaluate commonsense reasoning. Unlike the GSM8K-based datasets, which primarily require arithmetic reasoning, CSQA requires reasoning over implicit commonsense knowledge. We include CSQA to evaluate whether PLRM is effective beyond mathematical reasoning.

Out-of-Distribution Datasets. To evaluate out-of-distribution (OOD) generalization, we evaluate models trained on GSM8K-Aug-NL on three mathematical reasoning benchmarks that are not used during training.

SVAMP. SVAMP (Patel et al., 2021) is a collection of arithmetic word problems designed to test robustness to variations in problem structure and linguistic formulation. The benchmark provides a distribution shift from the GSM8K-based training data while retaining the requirement for multi-step arithmetic reasoning.

GSM-Hard. GSM-Hard (Gao et al., 2023) is a more challenging variant of GSM8K constructed by replacing numerical values with larger and less common numbers. This modification preserves the underlying reasoning structure while increasing the difficulty of numerical computation, allowing us to evaluate robustness under a controlled distribution shift.

MultiArith. MultiArith (Roy & Roth, 2015) consists of multi-step arithmetic word problems that require composing several arithmetic operations to obtain the final answer. We use MultiArith to assess whether reasoning learned from GSM8K-Aug-NL transfers to a distinct mathematical word-problem distribution.

B.2 *HiddenBranch* DATASET

HiddenBranch Construction. We construct *HiddenBranch* from the open-world abductive reasoning setting of ProofWriter (Tafjord et al., 2021), using examples from depths 1, 2, 3, and 5. We retain source questions that admit between two and five valid hidden facts, where each hidden fact independently makes the query provable when added to the original facts and rules. For each hidden fact, we use the corresponding ProofWriter proof as the ground-truth reasoning trajectory after verifying that the hidden fact independently entails the query.

A key property of *HiddenBranch* is that a single unique input can be associated with multiple valid reasoning–answer pairs. The input, consisting of the task instruction, facts, rules, and question, remains identical across branches, while each branch provides a different ground-truth reasoning trajectory and corresponding answer. During training and validation, these branches are represented as separate supervised examples sharing the same input. At test time, each unique input is presented only once and evaluated against the complete set of valid hidden answers.

Example. The following example illustrates two branches associated with the same unique input.

Input
Task: Find one missing fact that, when assumed by itself in addition to the given Facts and Rules, makes the Question provable. Give one proof for that missing fact.
Facts:
```
triple3. Harry is green.
triple2. Dave is rough.
triple1. Anne is cold.
```
Rules:
```
rule1. All cold, round people are green.
rule2. If Harry is round and Harry is not rough then Harry
is big.
rule3. If Anne is young and Anne is rough then Anne is big.
rule4. All green people are young.
rule5. Young, big people are kind.
rule6. If someone is cold then they are round.
```
Question:
```
Anne is kind.
```

For this same input, *HiddenBranch* contains multiple valid ground-truth reasoning–answer pairs:

GT Pair 1
GT-CoT:
```
B1: Assume Anne is big.
Anne is cold. --rule6--> Anne is round.;
Anne is cold. & Anne is round. --rule1--> Anne is green.;
Anne is green. --rule4--> Anne is young.;
Anne is young. & Anne is big. --rule5--> Anne is kind.
```
GT-Answer:
```
["Anne is big."]
```

GT Pair 2
GT-CoT:
```
B1: Assume Anne is rough.
Anne is cold. --rule6--> Anne is round.;
Anne is cold. & Anne is round. --rule1--> Anne is green.;
Anne is green. --rule4--> Anne is young.;
Anne is young. & Anne is rough. --rule3--> Anne is big.;
Anne is young. & Anne is big. --rule5--> Anne is kind.
```
GT-Answer:
```
["Anne is rough."]
```

Although the two examples share exactly the same input, they correspond to different valid hidden facts and therefore induce different reasoning trajectories and answers. This construction explicitly creates a one-to-many mapping from a unique input to multiple valid reasoning–answer pairs, which allows us to evaluate whether a stochastic latent reasoning model can represent and recover heterogeneous reasoning strategies for the same problem.

Table 7: Statistics of the *HiddenBranch* dataset. Each unique input corresponds to one source question, while valid branches denote distinct reasoning–answer pairs associated with that input.

Split	**# Unique Inputs**	**# Valid Branches**
Train	21,466	52,734
Validation	3,179	7,712
Test	6,128	15,101
Total	30,773	75,547

Evaluation Metrics. For the multi-trajectory setting, we evaluate both deterministic answer correctness and the distribution of stochastic generations. Let G_i denote the set of valid gold answers for input i.

Hit Rate. We first perform greedy decoding and measure whether the deterministically generated answer belongs to the set of valid answers:

$$\mathrm{HitRate} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\hat{a}_i^{\mathrm{det}} \in G_i\right]. \tag{28}$$

Thus, Hit Rate measures whether the model can recover at least one valid reasoning branch under deterministic decoding.

TV Score. To evaluate stochastic generations, we draw R samples for each input and compare their empirical answer distribution with an ideal distribution that assigns equal probability to all valid gold answers and zero probability to invalid generations. For an input with $|G_i| = G$ gold answers, let $n_{i,g}$ be the number of samples matching gold answer g, and let $n_{i,\mathrm{inv}} = R - \sum_{g=1}^{G} n_{i,g}$ denote the number of invalid samples. We define

$$P_i = \left[\frac{n_{i,1}}{R}, \ldots, \frac{n_{i,G}}{R}, \frac{n_{i,\mathrm{inv}}}{R}\right], \qquad Q_i = \left[\frac{1}{G}, \ldots, \frac{1}{G}, 0\right]. \tag{29}$$

The per-example score is the complement of the total variation distance,

$$S_i = 1 - \mathrm{TV}(P_i, Q_i) = 1 - \frac{1}{2}\sum_{c} |P_i(c) - Q_i(c)|, \tag{30}$$

and we report

$$\mathrm{TVScore} = \frac{1}{N}\sum_{i=1}^{N} S_i. \tag{31}$$

The score ranges from 0 to 1, with higher values indicating better agreement with the ideal distribution. It jointly penalizes invalid generations and concentration on only a subset of valid branches: the score is maximized when all generations are valid and are distributed as uniformly as possible across the gold answers.

C ADDITIONAL EXPERIMENTAL RESULTS

Overview. In this section, we provide additional experimental results and analyses of PLRM. We first present a thorough comparison with previously reported results on in-distribution benchmarks, covering verbalized, deterministic latent, and stochastic latent reasoning methods across both GPT-2 and Llama-3.2-1B-Instruct backbones. We then focus on the multi-trajectory setting and analyze how the learned prior captures multiple trajectory-specific posterior modes as the number of valid reasoning branches increases. Finally, we provide qualitative examples showing that individual strategy modes induce distinct valid reasoning trajectories and corresponding answers.

C.1 Comparison with Previously Reported Results

Table 8: Accuracy (%) of GPT-2 and Llama-3.2-1B-Instruct on in-distribution tasks.

Type	Method	GPT-2			Llama-3.2-1B		
		GSM8K-NL	GSM8K	CSQA	GSM8K-NL	GSM8K	CSQA
Verbalized	Explicit CoT	34.8	**44.1**	36.9	**54.1**	**61.6**	68.2
Deterministic	COCONUT	–	34.1	–	–	–	–
	CODI	35.3	43.7	**44.0**	49.7	55.6	74.0
	SIM-CoT	–	42.6	–	–	56.1	–
Stochastic	ReGuLaR	–	–	–	20.2	34.9	–
	CoLaR	–	–	–	–	40.1	–
	PLRM (Ours)	**36.30**	38.40	43.40	42.08	50.00	**74.28**

Table 8 complements our main in-distribution evaluation with performance reported in prior work. Overall, the comparison is consistent with the distinction between equation-dominant and verbalized reasoning discussed in Section 5.2. On GSM8K, where intermediate reasoning is largely represented through mathematical operations, PLRM remains competitive with deterministic latent reasoning methods. In contrast, PLRM shows stronger performance on tasks with more heavily verbalized reasoning traces, where the model must compress lengthy natural-language reasoning into a small number of latent steps while preserving the information necessary for answer prediction. Importantly, across all settings where previously reported stochastic baselines are available, PLRM consistently achieves higher accuracy than existing stochastic latent reasoning methods. This suggests that PLRM improves the effectiveness of stochastic latent reasoning without sacrificing the competitive performance observed in deterministic latent approaches.

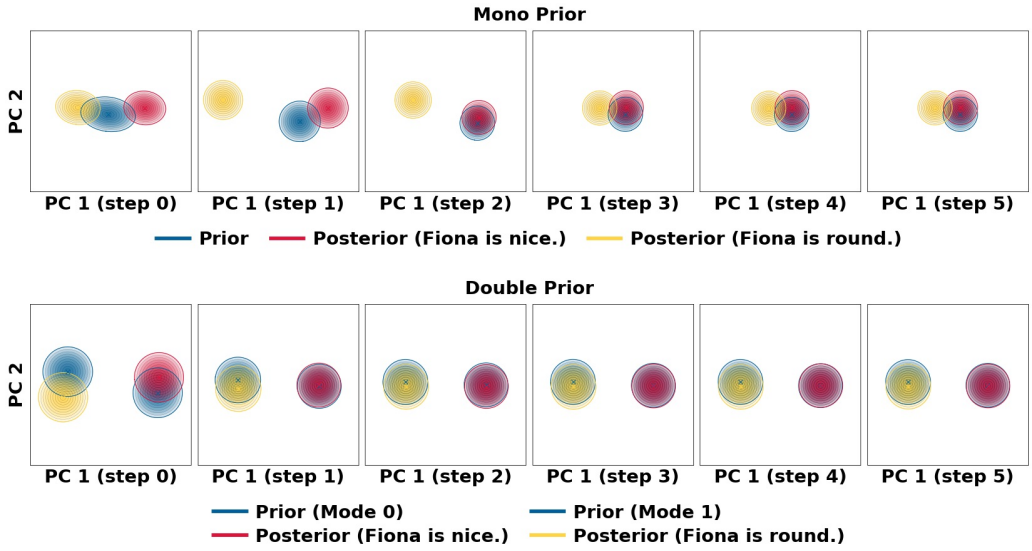

Figure 4: PCA visualization of prior and posterior latent distributions for a HiddenBranch example with two valid reasoning branches. Top: unimodal prior. Bottom: two-component strategy-conditioned prior.

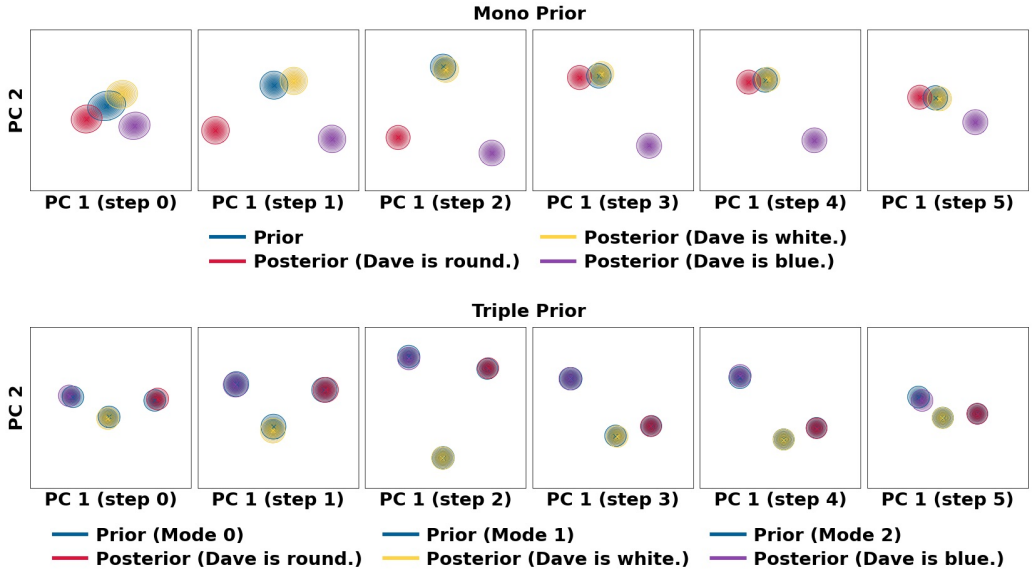

Figure 5: PCA visualization of prior and posterior latent distributions for a HiddenBranch example with three valid reasoning branches. Top: unimodal prior. Bottom: three-component strategy-conditioned prior.

C.2 Analysis of Heterogeneous Reasoning Strategies

Multimodal latent reasoning. To further examine how PLRM represents heterogeneous reasoning trajectories, we visualize the learned prior and posterior distributions in the latent space. Figures 4–6 show representative HiddenBranch examples with two, three, and four valid reasoning branches, respectively. For each example, we compare a unimodal prior with a strategy-conditioned

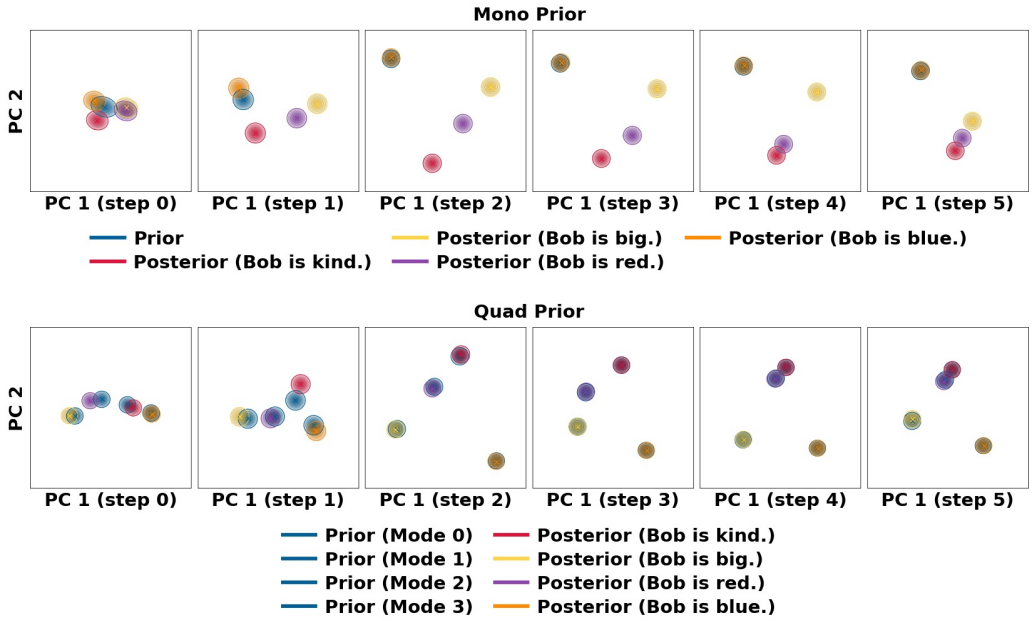

Figure 6: PCA visualization of prior and posterior latent distributions for a HiddenBranch example with four valid reasoning branches. Top: unimodal prior. Bottom: four-component strategy-conditioned prior.

prior whose number of components matches the number of valid reasoning branches. As shown in Figure 4, when two distinct posterior modes are induced by different valid reasoning trajectories, a single Gaussian prior must account for both modes simultaneously. In contrast, the strategy-conditioned prior maintains separate components corresponding to the different posterior modes across latent reasoning steps. Figures 5 and 6 show that the same behavior extends to settings with three and four valid reasoning branches. As the number of heterogeneous trajectories increases, the posterior exhibits multiple distinct modes, while the corresponding strategy-conditioned prior preserves multiple components rather than representing all trajectories with a single distribution. These visualizations are consistent with the motivation in Section 3.3: the discrete strategy variable separates heterogeneous reasoning modes, allowing the continuous latent variables to model variation within each mode.

Qualitative strategy-conditioned generations. To examine whether the separated latent modes correspond to meaningful differences in reasoning behavior, we further inspect generations conditioned on individual strategy modes. Figures 7–9 present representative HiddenBranch examples with two, three, and four valid reasoning branches, respectively. Given the same input, different strategy modes generate distinct valid missing facts together with their corresponding reasoning trajectories and answers. In the two-branch example (Figure 7), the two strategy modes recover different valid missing facts and follow different proof paths for the same query. Figures 8 and 9 extend this observation to three- and four-branch settings, where individual modes similarly generate distinct valid solutions. Importantly, the diversity across modes reflects differences in the underlying reasoning paths rather than merely variations in surface-level verbalization. Together with the latent-space visualizations above, these examples suggest that the learned strategy variable captures semantically distinct reasoning modes, while the continuous latent trajectory represents the reasoning process associated with each mode.

Task:
Find one missing fact that, when assumed by itself in addition to the given Facts and Rules, makes the Question provable.
Give one proof for that missing fact.

Facts:
triple4. Fiona is white.
triple1. Charlie is not blue.
triple6. Gary is rough.
triple7. Harry is big.
triple5. Gary is not cold.
triple2. Charlie is not rough.
triple3. Fiona is big.

Rules:
rule2. All rough, blue things are round.
rule3. If something is white and not rough then it is not round.
rule5. If something is round then it is nice.
rule4. If something is rough and not cold then it is not nice.
rule1. If Gary is rough and Gary is not big then Gary is nice.
rule6. If Fiona is nice then Fiona is blue.

Question:
Fiona is blue.

Mode 0

Explanation:
B1: Assume Fiona is round. Fiona is round. --rule5--> Fiona is nice.; Fiona is nice. --rule6--> Fiona is blue.

Answer:
Fiona is round.

GT CoT:
[branch 2 -- matches explanation's own assumption]
B2: Assume Fiona is round. Fiona is round. --rule5--> Fiona is nice.; Fiona is nice. --rule6--> Fiona is blue.

GT Answer:
Fiona is round.

Mode 1

Explanation:
B1: Assume Fiona is nice. Fiona is nice. --rule6--> Fiona is blue.

Answer:
Fiona is nice.

GT CoT:
[branch 1 -- matches explanation's own assumption]
B1: Assume Fiona is nice. Fiona is nice. --rule6--> Fiona is blue.

GT Answer:
Fiona is nice.

Figure 7: Qualitative HiddenBranch example with two valid reasoning branches generated from different strategy modes.

Task:
Find one missing fact that, when assumed by itself in addition to the given Facts and Rules, makes the Question provable.
Give one proof for that missing fact.

Facts:
triple2. Bob is furry.
triple6. Dave is not furry.
triple5. Bob is young.
triple7. Erin is white.
triple4. Bob is white.
triple1. Bob is blue.
triple8. Harry is not blue.
triple10. Harry is round.
triple3. Bob is smart.
triple9. Harry is red.

Rules:
rule5. If Erin is white and Erin is not round then Erin is not smart.
rule1. All furry, round things are blue.
rule3. White things are round.
rule7. If something is young and not blue then it is red.
rule4. If Dave is round then Dave is young.
rule6. If something is furry then it is red.
rule2. Blue things are white.
rule8. If Bob is not red then Bob is smart.

Question:
Dave is young.

Mode 0

Explanation:
B1: Assume Dave is white. Dave is white. --rule3--> Dave is round.; Dave is round. --rule4--> Dave is young.

Answer:
Dave is white.

GT CoT:
[branch 2 -- matches explanation's own assumption]
B2: Assume Dave is white. Dave is white. --rule3--> Dave is round.; Dave is round. --rule4--> Dave is young.

GT Answer:
Dave is white.

Mode 1

Explanation:
B1: Assume Dave is round. Dave is round. --rule4--> Dave is young.

Answer:
Dave is round.

GT CoT:
[branch 1 -- matches explanation's own assumption]
B1: Assume Dave is round. Dave is round. --rule4--> Dave is young.

GT Answer:
Dave is round.

Mode 2

Explanation:
B1: Assume Dave is blue. Dave is blue. --rule2--> Dave is white.; Dave is white. --rule3--> Dave is round.; Dave is round. --rule4--> Dave is young.

Answer:
Dave is blue.

GT CoT:
[branch 3 -- matches explanation's own assumption]
B3: Assume Dave is blue. Dave is blue. --rule2--> Dave is white.; Dave is white. --rule3--> Dave is round.; Dave is round. --rule4--> Dave is young.

GT Answer:
Dave is blue.

Figure 8: Qualitative HiddenBranch example with three valid reasoning branches generated from different strategy modes.

Task:
Find one missing fact that, when assumed by itself in addition to the given Facts and Rules, makes the Question provable.
Give one proof for that missing fact.

Facts:
triple4. Fiona is blue.
triple3. Dave is red.
triple1. Bob is white.
triple2. Charlie is white.

Rules:
rule3. If something is red then it is kind.
rule5. If something is big and green then it is red.
rule7. All blue, green things are red.
rule2. If something is blue then it is green.
rule8. If something is big then it is red.
rule1. If Charlie is blue and Charlie is green then Charlie is big.
rule6. If Fiona is nice then Fiona is green.
rule4. All kind, white things are green.

Question:
Bob is green.

Mode 0

Explanation:
B1: Assume Bob is red. Bob is red. --rule3--> Bob is kind.; Bob is kind. & Bob is white. --rule4--> Bob is green.

Answer:
Bob is red.

GT CoT:
[branch 3 -- matches explanation's own assumption]
B3: Assume Bob is red. Bob is red. --rule3--> Bob is kind.; Bob is kind. & Bob is white. --rule4--> Bob is green.

GT Answer:
Bob is red.

Mode 1

Explanation:
B1: Assume Bob is big. Bob is big. --rule8--> Bob is red.; Bob is red. --rule3--> Bob is kind.; Bob is kind. & Bob is red. --rule4--> Bob is green.

Answer:
Bob is big.

GT CoT:
[branch 2 -- matches explanation's own assumption]
B2: Assume Bob is big. Bob is big. --rule8--> Bob is red.; Bob is red. --rule3--> Bob is kind.; Bob is kind. & Bob is white. --rule4--> Bob is green.

GT Answer:
Bob is big.

Mode 2

Explanation:
B1: Assume Bob is blue. Bob is blue. --rule2--> Bob is green.

Answer:
Bob is blue.

GT CoT:
[branch 4 -- matches explanation's own assumption]
B4: Assume Bob is blue. Bob is blue. --rule2--> Bob is green.

GT Answer:
Bob is blue.

Mode 3

Explanation:
B1: Assume Bob is kind. Bob is kind. & Bob is white. --rule4--> Bob is green.

Answer:
Bob is kind.

GT CoT:
[branch 1 -- matches explanation's own assumption]
B1: Assume Bob is kind. Bob is kind. & Bob is white. --rule4--> Bob is green.

GT Answer:
Bob is kind.

Figure 9: Qualitative HiddenBranch example with four valid reasoning branches generated from different strategy modes.