

000
001
002
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053

VARIATIONAL REASONING FOR LANGUAGE MODELS

Anonymous authors

Paper under double-blind review

ABSTRACT

We introduce a **variational reasoning** framework for language models that treats thinking traces as latent variables and optimizes them through variational inference. Starting from the evidence lower bound (ELBO), we extend it to a multi-trace objective for tighter bounds and propose a forward-KL formulation that stabilizes the training of the variational posterior. We further show that rejection sampling finetuning and binary-reward RL, including GRPO, can be interpreted as local forward-KL objectives, where *an implicit weighting by model accuracy* naturally arises from the derivation and reveals a previously unnoticed bias toward easier questions. We empirically validate our method on the Qwen 2.5 and Qwen 3 model families across a wide range of reasoning tasks. Overall, our work provides a principled probabilistic perspective that unifies variational inference with RL-style methods and yields stable objectives for improving the reasoning ability of language models.

1 INTRODUCTION

Reasoning has recently become a central focus for large language models (LLMs), driving advances in tasks such as mathematics, coding, and scientific problem solving (Jaech et al., 2024; Comanici et al., 2025; Guo et al., 2025). A common strategy is to let models generate explicit thinking traces before producing final answers. To train such reasoning abilities, two dominant approaches are widely used: supervised finetuning (SFT) (Guha et al., 2025; Muennighoff et al., 2025) and reinforcement learning (RL) (Yu et al., 2025a; Liu et al., 2025; Zeng et al., 2025), both showing strong empirical success.

Despite this progress, each approach faces limitations. SFT often relies on curated long-thinking traces, which are costly to collect and, as an offline method, may struggle to generalize (Chu et al., 2025) or suffer from catastrophic forgetting (Shenfeld et al., 2025). Recent RL methods typically depend on verifiable rewards to mitigate reward hacking, yet training can be unstable and output diversity may collapse (Cheng et al., 2025; Cui et al., 2025b). As a result, correct answers to harder questions become increasingly rare, leading to lower Pass@K accuracy than base models (Yue et al., 2025a). These challenges motivate the search for a more principled objective for training reasoning models.

To this end, we propose to view reasoning through the lens of probabilistic modeling, where thinking traces are treated as *latent variables*. Variational inference (Kingma & Welling, 2013) provides a natural way to optimize the log-likelihood of producing correct answers. This perspective offers several advantages: it replaces the intractable marginalization over thinking traces with tractable lower bounds, enables multi-trace extensions that tighten the objective, and introduces a variational posterior that can sample thinking paths more likely to yield correct answers. In this way, it provides a principled objective for training reasoning models, while remaining compatible with verifiable rewards.

Building on this perspective, we develop a **variational reasoning** framework for language models in Section 2. The core idea is to decompose reasoning into a thinking trace and an answer, leading to the *maximum log-likelihood estimation (MLE)* objective. To make this optimization tractable, we introduce an evidence lower bound (ELBO) and extend it to an IWAE-style multi-trace formulation (Burda et al., 2015), which tightens with more rollouts. To further stabilize the training of the variational posterior, we propose a forward-KL objective that prevents collapse and makes better use of answer hints. Together, these components form a unified training pipeline (as shown in Algorithm 1) that jointly improves the reasoning model and the variational posterior.

Beyond the method itself, our framework also helps interpret existing approaches, as described in Section 3. Rejection-sampling finetuning (RFT) (Dong et al., 2023; Touvron et al., 2023) can be re-expressed as forward-KL optimization weighted by model accuracy, and binary-reward RL,

including GRPO (Shao et al., 2024), admits a similar form. Our analysis shows that this **weighting by accuracy** arises implicitly and produces a systematic bias toward easier questions, an effect that has not been explicitly recognized before. By placing these methods under a shared probabilistic view, our framework provides principled objectives and clarifies the behavior of widely used methods.

We validate our framework on the Qwen2.5 and Qwen3 model families (Yang et al., 2024; Team, 2025b) and observe consistent improvements over strong baselines across diverse reasoning benchmarks, including MATH500, AIME24&25, OlympiadBench, LiveCodeBench, GPQA-Diamond, and MMLU-Pro. Due to space constraints, a detailed discussion of related work is deferred to Appendix B.

2 VARIATIONAL REASONING

Let \mathcal{V}^* denote the set of all prompt strings over the vocabulary \mathcal{V} . Given an input question $\mathbf{x} \in \mathcal{V}^*$, a reasoning model $\pi_\theta(\mathbf{z}, \mathbf{y}|\mathbf{x})$ generates both a thinking process $\mathbf{z} \in \mathcal{V}^*$ and a predicted answer $\mathbf{y} \in \mathcal{V}^*$. The joint probability can be written as $\pi_\theta(\mathbf{z}, \mathbf{y}|\mathbf{x}) = \pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}) \cdot \pi_\theta(\mathbf{z}|\mathbf{x})$. Following a standard format template (Guo et al., 2025), these two conditional terms are computed as¹

$$\begin{aligned} \pi_\theta(\mathbf{z}|\mathbf{x}) &= \pi_\theta([\mathbf{z}, \text{</think>}] | [\mathbf{x}, \text{<think>}]), \\ \pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}) &= \pi_\theta([\mathbf{y}, \text{</answer>}] | [\mathbf{x}, \text{<think>}, \mathbf{z}, \text{</think>}, \text{<answer>}]), \end{aligned} \quad (1)$$

where </think> and </answer> serve as the end-of-sequence markers for \mathbf{z} and \mathbf{y} , respectively. We define the marginal distribution $P_\theta(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{z}} \pi_\theta(\mathbf{z}, \mathbf{y}|\mathbf{x}) = \sum_{\mathbf{z}} \pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}) \pi_\theta(\mathbf{z}|\mathbf{x})$, where the notation $P_\theta(\mathbf{y}|\mathbf{x})$ highlights that this distribution is *induced* by π_θ . This is different from the non-thinking probability $\pi_\theta(\mathbf{y}|\mathbf{x})$, which does not marginalize over possible thinking traces.

2.1 EVIDENCE LOWER BOUND

Let $\mathcal{Y}_x \subset \mathcal{V}^*$ denote the oracle set (possibly infinite) of correct answers to the question \mathbf{x} . The marginal probability that π_θ generates a correct answer is $P_\theta(\mathcal{Y}_x|\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}_x} P_\theta(\mathbf{y}|\mathbf{x})$. Maximizing this probability gives the *maximum log-likelihood estimation (MLE)* objective: $\max_\theta \log P_\theta(\mathcal{Y}_x|\mathbf{x})$. However, this MLE objective is intractable because computing $P_\theta(\mathbf{y}|\mathbf{x})$ requires summing over all possible thinking traces \mathbf{z} . To make learning feasible, we apply variational inference (Kingma & Welling, 2013) to derive an *evidence lower bound (ELBO)*:

$$\begin{aligned} \log P_\theta(\mathcal{Y}_x|\mathbf{x}) &= \log \sum_{\mathbf{z}} \pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}) \pi_\theta(\mathbf{z}|\mathbf{x}) \\ &= \log \mathbb{E}_{q_\phi(\mathbf{y}')} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')} \left[\frac{\pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}) \pi_\theta(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')} \right] \\ &\geq \mathbb{E}_{q_\phi(\mathbf{y}')} \underbrace{[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')} [\log \pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z})] - \mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}') || \pi_\theta(\mathbf{z}|\mathbf{x}))]}_{\mathcal{L}_{\text{ELBO}}(\mathbf{x}, \mathcal{Y}_x, \mathbf{y}'; \pi_\theta, q_\phi)}. \end{aligned} \quad (2)$$

In this expression, $\pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z})$ denotes the probability of producing a correct answer given the question \mathbf{x} and a particular thinking trace \mathbf{z} . The distribution $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')$ is the **variational posterior**, which conditions not only on the question \mathbf{x} but also on an auxiliary **answer hint \mathbf{y}'** :

$$q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}') = q_\phi([\mathbf{z}, \text{</think>}] | [\mathbf{x}, \text{<hint>}, \mathbf{y}', \text{</hint>}, \text{<think>}]). \quad (3)$$

Here, <hint> and </hint> are shown as example delimiters; in experiments, we ablate different special tokens to wrap the hint \mathbf{y}' and concatenate it after \mathbf{x} . Conditioning on \mathbf{y}' encourages the variational posterior to generate thinking traces \mathbf{z} that are more likely to yield correct answers. A simple yet effective design choice is to let \mathbf{y}' come directly from the oracle set, that is, $\text{supp}[q_\phi(\mathbf{y}')] \subset \mathcal{Y}_x$. In practice, \mathbf{y}' may be a rephrasing of a reference answer or any correct expression sampled from \mathcal{Y}_x .

We can further show (detailed in Appendix A.1) that maximizing the ELBO objective w.r.t. q_ϕ in Eq. (2) is equivalent to minimizing the *reverse KL* divergence between $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')$ and $P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_x)$:

$$\mathcal{L}_{\text{ELBO}}(\mathbf{x}, \mathcal{Y}_x, \mathbf{y}'; \pi_\theta, q_\phi) = \log P_\theta(\mathcal{Y}_x|\mathbf{x}) - \mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}') || P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_x)). \quad (4)$$

Here $P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_x) = \frac{\pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}) \pi_\theta(\mathbf{z}|\mathbf{x})}{P_\theta(\mathcal{Y}_x|\mathbf{x})}$ is the **true posterior**. Compared with the prior distribution $\pi_\theta(\mathbf{z}|\mathbf{x})$, this posterior distribution re-weights thinking traces by $\pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z})$, thus favoring \mathbf{z} that are more likely to produce correct answers. According to Eq. (4), we know that the optimal solution for $\max_{q_\phi} \mathcal{L}_{\text{ELBO}}(\mathbf{x}, \mathcal{Y}_x, \mathbf{y}'; \pi_\theta, q_\phi)$ is: $\forall \mathbf{y}' \sim q_\phi(\mathbf{y}')$, there is $q_\phi^*(\mathbf{z}|\mathbf{x}, \mathbf{y}') = P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_x)$.

¹We will omit special tokens such as </think> and </answer> in the formulas without ambiguity.

108 2.2 EXTENSION TO IWAE-STYLE LOWER BOUND
109

110 In reinforcement learning (RL), it is now common practice to perform parallel rollouts of multiple
111 thinking traces \mathbf{z} and answers \mathbf{y} for a given question \mathbf{x} (Shao et al., 2024). This naturally motivates
112 us to extend the single-trace ELBO in Eq. (4) to an importance-weighted autoencoder (IWAE) style
113 bound (Burda et al., 2015). By leveraging multiple K traces, this approach yields a strictly tighter
114 lower bound. Specifically, we obtain the following *IWAE-style lower bound* for $\log P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x})$:

$$115 \quad \mathcal{L}_{\text{ELBO}}^K(\mathbf{x}, \mathcal{Y}_\mathbf{x}, \mathbf{y}'; \pi_\theta, q_\phi) = \mathbb{E}_{\mathbf{z}_{1:K} \sim q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')} \left[\log \frac{1}{K} \sum_{k=1}^K \frac{\pi_\theta(\mathbf{z}_k, \mathcal{Y}_\mathbf{x}|\mathbf{x})}{q_\phi(\mathbf{z}_k|\mathbf{x}, \mathbf{y}')} \right]. \quad (5)$$

118 These IWAE-style bounds satisfy $\mathcal{L}_{\text{ELBO}}^K \leq \mathcal{L}_{\text{ELBO}}^{K+1} \leq \log P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x})$ for any $K \in \mathbb{N}^+$, which means
119 the bound becomes tighter as K increases (the proof is similar to that of Burda et al. (2015)). The
120 single-trace ELBO objective in Eq. (4) corresponds to the special case of $K = 1$, i.e., $\mathcal{L}_{\text{ELBO}} = \mathcal{L}_{\text{ELBO}}^1$.

121 **Gradient estimation.** We now derive the gradient of $\mathcal{L}_{\text{ELBO}}^K(\mathbf{x}, \mathcal{Y}_\mathbf{x}, \mathbf{y}'; \pi_\theta, q_\phi)$ w.r.t. the model
122 parameters θ (see Appendix A.2 for the gradient w.r.t. the variational parameters ϕ , i.e., $\nabla_\phi \mathcal{L}_{\text{ELBO}}^K$):
123

$$124 \quad \nabla_\theta \mathcal{L}_{\text{ELBO}}^K(\mathbf{x}, \mathcal{Y}_\mathbf{x}, \mathbf{y}'; \pi_\theta, q_\phi) = \mathbb{E}_{\mathbf{z}_{1:K} \sim q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')} \left[\sum_{k=1}^K \tilde{\rho}_k \nabla_\theta \log \pi_\theta(\mathbf{z}_k, \mathcal{Y}_\mathbf{x}|\mathbf{x}) \right], \quad (6)$$

$$125 \quad \text{where } \tilde{\rho}_k = \frac{\rho_k}{\sum_{j=1}^K \rho_j} \quad \text{and} \quad \rho_k = \frac{\pi_\theta(\mathbf{z}_k, \mathcal{Y}_\mathbf{x}|\mathbf{x})}{q_\phi(\mathbf{z}_k|\mathbf{x}, \mathbf{y}')},$$

126 **Estimating ρ_k .** The weight ρ_k in Eq. (6) can be decomposed as $\rho_k = \frac{\pi_\theta(\mathbf{z}_k|\mathbf{x})}{q_\phi(\mathbf{z}_k|\mathbf{x}, \mathbf{y}')} \cdot \pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}_k)$,
127 where the first term, $\frac{\pi_\theta(\mathbf{z}_k|\mathbf{x})}{q_\phi(\mathbf{z}_k|\mathbf{x}, \mathbf{y}')}$, is the *likelihood ratio of the thinking trace \mathbf{z}_k* , and the second term,
128 $\pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}_k)$, is the probability of producing a correct answer given \mathbf{x} and \mathbf{z}_k . In reasoning models,
129 a single trace \mathbf{z}_k may contain thousands of tokens. Directly computing the likelihood ratio over such
130 long sequences often leads to high variance, a phenomenon also reported in concurrent studies (Cetin
131 et al., 2025; Zheng et al., 2025). To mitigate this issue, we use the geometric mean $(\frac{\pi_\theta(\mathbf{z}_k|\mathbf{x})}{q_\phi(\mathbf{z}_k|\mathbf{x}, \mathbf{y}')})^{1/|\mathbf{z}_k|}$
132 as a surrogate for the likelihood ratio of \mathbf{z}_k . This per-token normalization reduces variance at the cost
133 of introducing some bias, effectively spreading the ratio evenly across the thinking tokens.
134

135 As for computing $\pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}_k)$, we consider two unbiased estimators: **(i) likelihood-based estimator**
136 is $\pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}) = |\mathcal{Y}_\mathbf{x}| \cdot \mathbb{E}_{\mathbf{y} \sim \mathcal{U}(\mathcal{Y}_\mathbf{x})} [\pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})]$, where $|\mathcal{Y}_\mathbf{x}|$ is cardinality of $\mathcal{Y}_\mathbf{x}$ and $\mathcal{U}(\mathcal{Y}_\mathbf{x})$ is the uniform
137 distribution on $\mathcal{Y}_\mathbf{x}$; **(ii) accuracy-based estimator** is $\pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}) = \mathbb{E}_{\mathbf{y} \sim \pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_\mathbf{x})]$,
138 where $\mathbb{1}(\cdot)$ is the indicator function. When $|\mathcal{Y}_\mathbf{x}| = 1$, i.e., there is a unique correct answer expression
139 \mathbf{y}^* , Zhou et al. (2025) show that the likelihood-based estimator has lower variance (in fact, zero)
140 compared to the accuracy-based one. We now extend this comparison to general cases when $|\mathcal{Y}_\mathbf{x}| > 1$:
141

142 **Theorem 1.** *(Proof in Appendix A.3) For $|\mathcal{Y}_\mathbf{x}| > 1$, the worst-case variances of the likelihood-based
143 estimator and the accuracy-based estimator over all possible π_θ (under fixed $\pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z})$) are*

$$144 \quad \max_{\pi_\theta} \text{Var}_{\text{like}} = (|\mathcal{Y}_\mathbf{x}| - 1) \cdot \pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z})^2; \quad \max_{\pi_\theta} \text{Var}_{\text{acc}} = \pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}) \cdot (1 - \pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z})). \quad (7)$$

145 *Therefore, the accuracy-based estimator has lower worst-case variance, i.e., $\max_{\pi_\theta} \text{Var}_{\text{acc}} \leq \max_{\pi_\theta} \text{Var}_{\text{like}}$, whenever the model accuracy (conditional on \mathbf{x}, \mathbf{z}) satisfies $\pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}) \geq \frac{1}{|\mathcal{Y}_\mathbf{x}|}$.*

146 Note that for many practical questions, the space of correct answers can be quite flexible, so typically
147 $|\mathcal{Y}_\mathbf{x}| \gg 1$. In this regime, the accuracy-based estimator enjoys much lower worst-case variance.
148 Based on this insight, in our experiments we estimate the weight ρ_k as
149

$$150 \quad \rho_k^{\text{est}} = \left(\frac{\pi_\theta(\mathbf{z}_k|\mathbf{x})}{q_\phi(\mathbf{z}_k|\mathbf{x}, \mathbf{y}')} \right)^{1/|\mathbf{z}_k|} \cdot \mathbb{E}_{\mathbf{y} \sim \pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}_k)} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_\mathbf{x})], \quad (8)$$

151 where the expectation $\mathbb{E}_{\mathbf{y} \sim \pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}_k)} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_\mathbf{x})]$ is approximated by sampling multiple candidate
152 answers for each thinking trace \mathbf{z}_k , similar to the implementation in Qi et al. (2025).

153 **Estimating $\nabla_\theta \log \pi_\theta(\mathbf{z}_k, \mathcal{Y}_\mathbf{x}|\mathbf{x})$.** When evaluating $\nabla_\theta \mathcal{L}_{\text{ELBO}}^K$ in Eq. (6), we need the gradient
154 $\nabla_\theta \log \pi_\theta(\mathbf{z}_k, \mathcal{Y}_\mathbf{x}|\mathbf{x}) = \nabla_\theta \log \pi_\theta(\mathbf{z}_k|\mathbf{x}) + \nabla_\theta \log \pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}_k)$. The first term, $\nabla_\theta \log \pi_\theta(\mathbf{z}_k|\mathbf{x})$,
155 is straightforward to calculate. For the second term, $\nabla_\theta \log \pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}_k)$, we also adopt an *accuracy-based estimator*:
156 $\nabla_\theta \log \pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}_k) = \frac{\mathbb{E}_{\mathbf{y} \sim \pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}_k)} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_\mathbf{x}) \nabla_\theta \log \pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}_k)]}{\mathbb{E}_{\mathbf{y} \sim \pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}_k)} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_\mathbf{x})]}$. In practice, the
157 expectations w.r.t. $\pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}_k)$ are approximated using the same samples drawn to estimate ρ_k^{est} .
158

162 **Algorithm 1** Training pipeline of variational reasoning

163
Inputs: An initial reasoning model $\pi_{\theta_0}(\mathbf{z}, \mathbf{y}|\mathbf{x})$, variational posterior $q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')$, question-answer
164 dataset $\{\mathbf{x}, \mathbf{y}_x^*\} \in \mathcal{X}$, where $\mathbf{y}_x^* \in \mathcal{Y}_x$ is one of the reference answers corresponding to \mathbf{x}
165
Inputs: Rollout numbers K and M , training rounds T , steps per round S_{θ} and S_{ϕ} , optimizer \mathcal{O}
166
Outputs: The trained model parameters θ_T and variational parameters ϕ_T

167 1: **Initialize** $q_{\phi_0}(\mathbf{z}|\mathbf{x}, \mathbf{y}') \xleftarrow{\phi_0 \text{ copy } \theta_0} \pi_{\theta_0}([\mathbf{z}, \text{</think>}][\mathbf{x}, \text{</hint>}, \mathbf{y}', \text{</hint>}, \text{<think>}])$
168 2: **Construct** \mathcal{Y}_x (or its subset by rephrasing \mathbf{y}_x^*) and rule-based/model-based verifier $\mathbb{1}(\mathbf{y} \in \mathcal{Y}_x)$
169 3: **for** $t = 1$ to T **do**
170 ## Updating variational parameters ϕ_t with $\nabla_{\phi} \mathcal{L}_{\text{forward}}^M$ in Eq. (9); initializing $\phi_t \leftarrow \phi_{t-1}$
171 4: **for** $s = 1$ to S_{ϕ} **do**
172 5: Sample a training batch of questions $\mathcal{B} \subset \mathcal{X}$
173 6: **for all** questions $\mathbf{x} \in \mathcal{B}$ **do** ## Collecting $\mathbf{z}_{1:M}$ and compute weights \tilde{w}_m for each m
174 7: **Rollout** $\mathbf{z}_{1:M} \sim \pi_{\theta_{t-1}}(\mathbf{z}|\mathbf{x})$, $\mathbf{y}' \sim q_{\phi}(\mathbf{y}') = \mathcal{U}(\mathcal{Y}_x)$
175 8: **Compute** $w_m = \mathbb{E}_{\mathbf{y} \sim \pi_{\theta_{t-1}}(\mathbf{y}|\mathbf{x}, \mathbf{z}_m)} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_x)]$ and $\tilde{w}_m = \frac{w_m}{\sum_{j=1}^M w_j}$
176 9: **Update** $\phi_t \leftarrow \mathcal{O}.\text{step} \left(\phi_t, \left[\frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} \sum_{m=1}^M \tilde{w}_m \nabla_{\phi_t} \log q_{\phi_t}(\mathbf{z}_m|\mathbf{x}, \mathbf{y}') \right] \right)$
177 ## Updating model parameters θ_t with $\nabla_{\theta} \mathcal{L}_{\text{ELBO}}^K$ in Eq. (6); initializing $\theta_t \leftarrow \theta_{t-1}$
178 10: **for** $s = 1$ to S_{θ} **do**
179 11: Sample a training batch of questions $\mathcal{B} \subset \mathcal{X}$
180 12: **for all** questions $\mathbf{x} \in \mathcal{B}$ **do** ## Collecting $\mathbf{z}_{1:K}$ and compute weights $\tilde{\rho}_k$ for each k
181 13: **Rollout** $\mathbf{z}_{1:K} \sim q_{\phi_t}(\mathbf{z}|\mathbf{x}, \mathbf{y}')$, $\mathbf{y}' \sim q_{\phi}(\mathbf{y}') = \mathcal{U}(\mathcal{Y}_x)$ ## Estimate ρ_k^{est} by Eq. (8)
182 14: **Compute** $\rho_k^{\text{est}} = \left(\frac{\pi_{\theta_t}(\mathbf{z}_k|\mathbf{x})}{q_{\phi_t}(\mathbf{z}_k|\mathbf{x}, \mathbf{y}')} \right)^{1/|\mathbf{z}_k|} \cdot \mathbb{E}_{\mathbf{y} \sim \pi_{\theta_t}(\mathbf{y}|\mathbf{x}, \mathbf{z}_k)} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_x)]$ and $\tilde{\rho}_k = \frac{\rho_k^{\text{est}}}{\sum_{j=1}^K \rho_j^{\text{est}}}$
183 15: **Compute** $\nabla_{\theta_t} \log \pi_{\theta_t}(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}_k) = \frac{\mathbb{E}_{\mathbf{y} \sim \pi_{\theta_t}(\mathbf{y}|\mathbf{x}, \mathbf{z}_k)} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_x) \nabla_{\theta_t} \log \pi_{\theta_t}(\mathbf{y}|\mathbf{x}, \mathbf{z}_k)]}{\mathbb{E}_{\mathbf{y} \sim \pi_{\theta_t}(\mathbf{y}|\mathbf{x}, \mathbf{z}_k)} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_x)]}$
184 16: **Update** $\theta_t \leftarrow \mathcal{O}.\text{step} \left(\theta_t, \left[\frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} \sum_{k=1}^K \tilde{\rho}_k \nabla_{\theta_t} (\log \pi_{\theta_t}(\mathbf{z}_k|\mathbf{x}) + \log \pi_{\theta_t}(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}_k)) \right] \right)$
185 17: **return** θ_T and ϕ_T

191 2.3 OPTIMIZING THE VARIATIONAL POSTERIOR VIA FORWARD KL DIVERGENCE
192

193 While Eq. (5) provides IWAE-style bounds that yield tighter optimization of the MLE objective w.r.t.
194 the model parameters θ (through $\nabla_{\theta} \mathcal{L}_{\text{ELBO}}^K$), our pilot experiments show unexpected behavior for the
195 optimization of the variational parameters ϕ (through $\nabla_{\phi} \mathcal{L}_{\text{ELBO}}$ or $\nabla_{\phi} \mathcal{L}_{\text{ELBO}}^K$). Recall from Eq. (4)
196 that the ELBO objective minimizes the *reverse* KL divergence $\mathbb{D}_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')||P_{\theta}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_x))$,
197 where both the expectation and Monte Carlo samples are taken under $q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')$. In practice,
198 however, the policy model $\pi_{\theta}(\mathbf{z}|\mathbf{x})$ is often already well-trained due to pretraining of base LLMs (Liu
199 et al., 2025), while the variational posterior $q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')$ may *struggle to effectively use hints \mathbf{y}' from*
200 *correct answers without collapsing into shortcut reasoning* (e.g., directly leaking answer tokens into
201 the thinking trace). To address this imbalance, we propose to optimize $q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')$ using the *forward*
202 KL divergence $\mathbb{D}_{\text{KL}}(P_{\theta}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_x)||q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}'))$, whose gradient w.r.t. ϕ can be written as:²

203
$$\nabla_{\phi} \mathbb{D}_{\text{KL}}(P_{\theta}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_x)||q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')) \simeq \mathbb{E}_{\mathbf{z}_{1:M} \sim \pi_{\theta}(\mathbf{z}|\mathbf{x})} \left[\sum_{m=1}^M \tilde{w}_m \nabla_{\phi} \log q_{\phi}(\mathbf{z}_m|\mathbf{x}, \mathbf{y}') \right] \triangleq \nabla_{\phi} \mathcal{L}_{\text{forward}}^M, \quad (9)$$

204 205 where $\tilde{w}_m = \frac{w_m}{\sum_{j=1}^M w_j}$ and $w_m = \pi_{\theta}(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}_m) = \mathbb{E}_{\mathbf{y} \sim \pi_{\theta}(\mathbf{y}|\mathbf{x}, \mathbf{z}_m)} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_x)]$.

206 This approximation, $\nabla_{\phi} \mathcal{L}_{\text{forward}}^M$, follows a derivation similar to Bornschein & Bengio (2015) (see
207 Appendix A.4), with the sample size M not necessarily equal to K used in $\nabla_{\phi} \mathcal{L}_{\text{ELBO}}^K$. Unlike IWAE,
208 this objective is an approximation rather than a lower bound. Optimizing Eq. (9) can be viewed as a
209 weighted supervised finetuning (SFT) for $q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')$, where training data is sampled from $\pi_{\theta}(\mathbf{z}|\mathbf{x})$.

210 We summarize the overall training pipeline of our variational reasoning method in Algorithm 1. In our
211 experiments, we train for only a single round ($T = 1$), leaving multi-round training as an interesting
212 direction for future work.

213 ²We assume that $q_{\phi}(\mathbf{y}')$ is a fixed prior distribution and does not involve gradients.
214

216 **3 CONNECTION TO OTHER METHODS**

218 In prior work, the thinking trace \mathbf{z} and the final answer \mathbf{y} are often treated together as the full output
 219 in the formulations (Shao et al., 2024; Guo et al., 2025; Liu et al., 2025; Wu et al., 2025). By explicitly
 220 decomposing the output into a thinking process \mathbf{z} and an answer \mathbf{y} , as we have done above, we can
 221 gain fresh perspectives on how our formulation relates to other mainstream methods.

222 **Connection to rejection sampling finetuning (RFT).** RFT methods (Dong et al., 2023; Touvron
 223 et al., 2023) generate multiple candidate outputs for each input \mathbf{x} using a reference model π_{ref} , and
 224 then select the candidate with the highest reward. The reference model may be a strong teacher model
 225 or identical to the learner π_θ (i.e., $\pi_{\text{ref}} = \pi_\theta^{\text{sg}}$, with sg denoting stop-gradient). Formally, the gradient
 226 of RFT training objective, *focusing only on the learning of the thinking trace \mathbf{z}* , can be written as:

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\text{RFT}}(\mathbf{x}, \pi_\theta) &\triangleq \nabla_\theta \mathbb{E}_{\pi_{\text{ref}}(\mathbf{z}|\mathbf{x})} \mathbb{E}_{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x}, \mathbf{z})} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_\mathbf{x}) \cdot \log \pi_\theta(\mathbf{z}, \mathbf{y}|\mathbf{x})] \\ &= \nabla_\theta \mathbb{E}_{\pi_{\text{ref}}(\mathbf{z}|\mathbf{x})} [\pi_{\text{ref}}(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}) \cdot (\log \pi_\theta(\mathbf{z}|\mathbf{x}) + \log \pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}))] \\ &\xrightarrow[\pi_\theta(\mathbf{z}|\mathbf{x})]{\text{only w.r.t.}} \nabla_\theta \mathbb{E}_{\pi_{\text{ref}}(\mathbf{z}|\mathbf{x})} [\pi_{\text{ref}}(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}) \cdot \log \pi_\theta(\mathbf{z}|\mathbf{x})] \\ &= -P_{\text{ref}}(\mathcal{Y}_\mathbf{x}|\mathbf{x}) \cdot \nabla_\theta \mathbb{D}_{\text{KL}}(P_{\text{ref}}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x}) \parallel \pi_\theta(\mathbf{z}|\mathbf{x})). \end{aligned} \quad (10)$$

233 Here $P_{\text{ref}}(\mathcal{Y}_\mathbf{x}|\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}_\mathbf{x}} P_{\text{ref}}(\mathbf{y}|\mathbf{x})$ denotes the *model accuracy* on instruction \mathbf{x} , and the true
 234 posterior of the reference model is $P_{\text{ref}}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x}) = \frac{\pi_{\text{ref}}(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}) \pi_{\text{ref}}(\mathbf{z}|\mathbf{x})}{P_{\text{ref}}(\mathcal{Y}_\mathbf{x}|\mathbf{x})}$. As seen, the RFT objective
 235 can be viewed as maximizing a forward KL divergence weighted by $P_{\text{ref}}(\mathcal{Y}_\mathbf{x}|\mathbf{x})$, with the optimal
 236 solution $\pi_\theta^*(\mathbf{z}|\mathbf{x}) = P_{\text{ref}}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x})$. In practice, this weighting downplays hard questions with small
 237 $P_{\text{ref}}(\mathcal{Y}_\mathbf{x}|\mathbf{x})$, biasing training toward easier ones. In contrast, our formulation in Eq. (9) treats all
 238 questions more evenly, ensuring that the objective remains attentive to difficult cases.

239 **Connection to binary reward RL.** In the case of RL training with a 0–1 binary reward, the training
 240 objective can be written in a form similar to Eq. (10), focusing only on the thinking trace \mathbf{z} :

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\text{bi-RL}}(\mathbf{x}, \pi_\theta) &\triangleq \nabla_\theta \mathbb{E}_{\pi_\theta(\mathbf{z}|\mathbf{x})} \mathbb{E}_{\pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_\mathbf{x})] \\ &\xrightarrow[\pi_\theta(\mathbf{z}|\mathbf{x})]{\text{only w.r.t.}} \nabla_\theta \mathbb{E}_{\pi_\theta^{\text{sg}}(\mathbf{z}|\mathbf{x})} [\pi_\theta^{\text{sg}}(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}) \cdot \log \pi_\theta(\mathbf{z}|\mathbf{x})] \\ &= -P_\theta^{\text{sg}}(\mathcal{Y}_\mathbf{x}|\mathbf{x}) \cdot \nabla_\theta \mathbb{D}_{\text{KL}}(P_\theta^{\text{sg}}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x}) \parallel \pi_\theta(\mathbf{z}|\mathbf{x})). \end{aligned} \quad (11)$$

246 Thus, the local gradient for $\pi_\theta(\mathbf{z}|\mathbf{x})$ in binary-reward RL is equivalent to minimizing the forward
 247 KL divergence between $\pi_\theta(\mathbf{z}|\mathbf{x})$ and the true posterior $P_\theta^{\text{sg}}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x}) = \frac{\pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}) \pi_\theta(\mathbf{z}|\mathbf{x})}{P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x})}$, with the
 248 update further weighted by the model accuracy $P_\theta^{\text{sg}}(\mathcal{Y}_\mathbf{x}|\mathbf{x})$. Interestingly, the RL training objective
 249 itself is $\mathcal{L}_{\text{bi-RL}}(\mathbf{x}, \pi_\theta) = \mathbb{E}_{\pi_\theta(\mathbf{z}|\mathbf{x})} \mathbb{E}_{\pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_\mathbf{x})] = P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x})$, whereas the MLE objective
 250 in Eq. (2) for our variational reasoning framework instead maximizes $\log P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x})$.

252 Furthermore, in Group Relative Policy Optimization (**GRPO**) (Shao et al., 2024), one of the most
 253 widely used RL training objectives, the reward is normalized by the standard deviation of rewards
 254 within a group of rollouts. Under 0–1 binary reward, each rollout reward follows a Bernoulli distri-
 255 bution with mean $P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x})$ and standard deviation $\sqrt{P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}) \cdot (1 - P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}))}$. The gradient of
 256 the GRPO objective can therefore be derived as (we omit the min and clip operations for brevity)

$$\nabla_\theta \mathcal{L}_{\text{bi-GRPO}}(\mathbf{x}, \pi_\theta) \xrightarrow[\pi_\theta(\mathbf{z}|\mathbf{x})]{\text{only w.r.t.}} -\sqrt{\frac{P_\theta^{\text{sg}}(\mathcal{Y}_\mathbf{x}|\mathbf{x})}{1 - P_\theta^{\text{sg}}(\mathcal{Y}_\mathbf{x}|\mathbf{x})}} \cdot \nabla_\theta \mathbb{D}_{\text{KL}}(P_\theta^{\text{sg}}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x}) \parallel \pi_\theta(\mathbf{z}|\mathbf{x})), \quad (12)$$

260 which follows from $\nabla_\theta \mathcal{L}_{\text{bi-GRPO}}(\mathbf{x}, \pi_\theta) = \nabla_\theta \mathcal{L}_{\text{bi-RL}}(\mathbf{x}, \pi_\theta) / \sqrt{P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}) \cdot (1 - P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}))}$. Thus,
 261 the local gradient for $\pi_\theta(\mathbf{z}|\mathbf{x})$ in GRPO is still equivalent to minimizing the forward KL divergence
 262 between $\pi_\theta(\mathbf{z}|\mathbf{x})$ and the true posterior $P_\theta^{\text{sg}}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x})$. Note that the per-instruction weight becomes
 263 $\sqrt{P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}) / (1 - P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}))}$, which increases monotonically with model accuracy and therefore
 264 also emphasizes easier questions with higher $P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x})$. In Appendix A.5, we extend these analyses
 265 and derive gradients for more general RL reward shaping, including cases with a *format reward*.

266 **4 EXPERIMENTS**

268 **Datasets.** We train on the Bespoke-Stratos-17k dataset curated by Li et al. (2025a), which combines
 269 math problems from Numina-Math (Li et al., 2024) with code problems from APPS (Hendrycks
 et al., 2021a) and TACO (Li et al., 2023). The dataset contains 16,710 samples, each paired with

Table 1: Performance of models trained from **Qwen3-4B-Base**. All models are trained on Bespoke-Stratos-17k except for General-Reasoner-4B. The best and second-best results are highlighted using **bold text** and underlined text, respectively.

Method	MATH500	AIME24	AIME25	AMC23	OlympiadBench	Avg
	Avg@2	Avg@32	Avg@32	Avg@32	Avg@2	
Qwen3-4B-Base	45.30	4.79	5.73	27.73	23.37	21.38
General-Reasoner-4B	71.70	19.06	16.77	55.00	45.18	41.54
Bespoke-Stratos-4B [†]	84.70	27.29	24.17	70.16	50.45	51.35
Ours-PB-GML-4B	87.30	33.54	<u>26.77</u>	<u>74.06</u>	<u>54.45</u>	<u>55.23</u>
Ours-PB-Acc-4B	88.30	<u>31.67</u>	27.29	75.63	55.71	55.72

Method	GPQA-D	LCB-E	LCB-M	LCB-H	MMLU-Pro	Avg
	Avg@8	Avg@8	Avg@8	Avg@8	Avg@1	
Qwen3-4B-Base	29.10	18.54	5.46	1.32	36.89	18.26
General-Reasoner-4B	40.97	61.40	17.90	2.85	61.36	36.90
Bespoke-Stratos-4B [†]	44.95	71.22	19.54	3.25	63.03	40.40
Ours-PB-GML-4B	45.52	<u>79.53</u>	<u>31.25</u>	6.20	<u>65.52</u>	<u>45.60</u>
Ours-PB-Acc-4B	<u>45.33</u>	80.29	33.68	<u>5.79</u>	65.53	46.12

Table 2: Performance of models trained from **Qwen3-8B-Base** trained on Bespoke-Stratos-17k.

Method	MATH500	AIME24	AIME25	AMC23	OlympiadBench	Avg
	Avg@2	Avg@32	Avg@32	Avg@32	Avg@2	
Qwen3-8B-Base	65.20	11.46	10.10	45.00	34.72	33.30
Bespoke-Stratos-8B [†]	89.70	39.58	28.85	78.91	55.64	58.54
Ours-PB-GML-8B	<u>91.60</u>	<u>44.06</u>	<u>31.67</u>	<u>83.59</u>	<u>58.23</u>	<u>61.83</u>
Ours-PB-Acc-8B	91.80	45.63	31.98	85.47	58.98	62.77

Method	GPQA-D	LCB-E	LCB-M	LCB-H	MMLU-Pro	Avg
	Avg@8	Avg@8	Avg@8	Avg@8	Avg@1	
Qwen3-8B-Base	35.42	41.14	13.65	1.42	45.62	27.45
Bespoke-Stratos-8B [†]	<u>53.03</u>	81.53	36.89	7.11	<u>68.74</u>	49.46
Ours-PB-GML-8B	52.72	87.36	<u>45.51</u>	13.82	70.76	<u>54.03</u>
Ours-PB-Acc-8B	53.66	86.47	49.33	13.21	70.76	54.69

a long-thinking trace generated by DeepSeek-R1 (Guo et al., 2025) or QwQ-32B-Preview (Team, 2024). To ensure fair evaluation, all training data is strictly separated from the test sets.

Baselines. We compare against a broad set of baselines for rigorous evaluation. For Qwen2.5-Instruct models, we include Bespoke-Stratos (Labs, 2025) and RLT (Cetin et al., 2025) at the 7B and 32B scales, where RLT trains teachers via RL to improve distillation. For Qwen3-Base models, we compare with Bespoke-Stratos-4B/8B[†] and General-Reasoner-4B (Ma et al., 2025), the latter trained with GRPO (Shao et al., 2024) and a model-based verifier. Models marked [†] are trained by us; others are official releases. All Bespoke-Stratos models are distilled on Bespoke-Stratos-17k. This ensures fair comparison, as all methods (except General-Reasoner) follow the same training recipes and datasets.

Evaluation. We assess our models on a broad set of challenging benchmarks: **MATH500** (Hendrycks et al., 2021b), a subset of competition math curated by Lightman et al. (2024); **AIME24&25** (MAA, 2025); **AMC23** (MAA, 2023); **OlympiadBench** (He et al., 2024); **LiveCodeBench** (Jain et al., 2025), with Easy, Medium, and Hard subsets (**LCB-E**, **LCB-M**, **LCB-H**) for fine-grained coding evaluation; **GPQA-Diamond** (Rein et al., 2024) (**GPQA-D**), graduate-level natural science questions; and **MMLU-Pro** (Wang et al., 2024), a diverse multiple-choice benchmark. Among these, GPQA-D and MMLU-Pro are out-of-distribution (OOD) relative to our training data.

Decoding and average accuracy. Following Hochlehnert et al. (2025), we sample responses with `temperature=0.7` and report average accuracy over k responses per question (Avg@ k). To reduce randomness and ensure fair comparison, we use larger k for smaller datasets: Avg@32 for AIME24, AIME25, and AMC23; Avg@8 for GPQA-Diamond, LCB-E, LCB-M, and LCB-H; Avg@2 for MATH500 and OlympiadBench; and Avg@1 for the large MMLU-Pro (12k+ questions). All evaluations are conducted with SkyThought (Team, 2025a), with additional details in Appendix D.

Table 3: Performance of models trained from Qwen2.5-32B-Instruct trained on Bespoke-Stratos-17k. The best and second-best results are highlighted using **bold text** and underlined text, respectively.

Method	MATH500 Avg@2	AIME24 Avg@32	AIME25 Avg@32	AMC23 Avg@32	OlympiadBench Avg@2	Avg
Qwen2.5-32B-Instruct	80.70	15.83	12.08	61.95	46.96	43.51
Bespoke-Stratos-32B	92.60	55.42	46.88	92.19	64.68	70.34
RLT-32B	93.50	56.77	47.19	91.48	63.21	70.43
Ours-PA-GML-32B	<u>93.20</u>	<u>56.56</u>	<u>48.13</u>	93.98	64.24	<u>71.22</u>
Ours-PA-Acc-32B	93.50	58.85	50.31	<u>92.97</u>	<u>64.39</u>	72.01

Method	GPQA-D Avg@8	LCB-E Avg@8	LCB-M Avg@8	LCB-H Avg@8	MMLU-Pro Avg@1	Avg
Qwen2.5-32B-Instruct	46.28	79.88	40.60	9.76	59.19	47.14
Bespoke-Stratos-32B	57.57	<u>94.78</u>	73.54	30.48	75.22	66.32
RLT-32B	59.09	93.20	72.15	29.78	74.88	65.82
Ours-PA-GML-32B	60.92	95.19	72.21	35.57	75.57	67.89
Ours-PA-Acc-32B	60.73	94.78	73.18	31.81	75.55	67.21

Figure 1: Training loss and gradient norm of different methods during Qwen3-Base model training.

4.1 TRAINING DETAILS FOR VARIATIONAL REASONING

We conduct experiments on Qwen2.5-7B-Instruct, Qwen2.5-32B-Instruct (Yang et al., 2024), Qwen3-4B-Base, and Qwen3-8B-Base (Team, 2025b). Following Algorithm 1, we first train an *initial* reasoning model π_{θ_0} on Bespoke-Stratos-17k using the recipe from Labs (2025), and then a variational posterior q_ϕ with the forward KL divergence (Eq. (9)) on the same dataset. These models are later used to compute the weights $\tilde{\rho}_k$. For π_{θ_0} , we adopt the prompt template from Labs (2025); for q_ϕ , we test two alternative templates (“-PA” and “-PB”, see Appendix E). Both π_{θ_0} and q_ϕ are finetuned independently from the same base model without weight sharing (Appendix C.1).

Next, we use the trained q_ϕ to generate 8 responses (thinking traces and final answers) per training sample. For each response, the weight $\tilde{\rho}_k$ in Eq. (6) is computed from q_ϕ , π_{θ_0} , and, when using the accuracy-based estimator, math/code verifiers from SkyThought. To estimate $\pi_\theta(\mathcal{Y}_x | \mathcal{X}, \mathcal{Z})$ in $\tilde{\rho}_k$, we compare three options: a naive likelihood method (“-L”), an accuracy-based method (“-Acc”, Section 2.2), and a geometric mean of token-level probabilities (“-GML”), as detailed in Appendix C.2.

We train the final reasoning model π_θ following Eq. (6) under two data settings. **17K**: the full Bespoke-Stratos-17k dataset. To enhance efficiency, we create a mixed dataset containing, for each original sample, the q_ϕ -generated response with the highest $\tilde{\rho}_k$ and the original sample itself. **1K**: a fixed 1,000-sample subset uniformly drawn from the full dataset, where all 8 q_ϕ -generated responses per sample are used for weighted SFT with $\tilde{\rho}_k$. The same 1K subset is reused across related experiments. Main results are reported with 17K, while ablations use both 17K and 1K configurations (Appendix C.3).

4.2 MAIN RESULTS

We evaluate our method across four model variants: Qwen3-4B/8B-Base (Tables 1 and 2) and Qwen2.5-7B/32B-Instruct (Tables 3 and 6). Extended results are provided in Appendix F.1.

Variational reasoning performance. All methods substantially improve the reasoning ability of the base model, but our approach consistently achieves the best results. As shown in Tables 1 and 2, variational reasoning yields substantial improvements in math, code, and other general domains compared to the base model (e.g., over 160% improvement in math and over 152% in other domains). It also surpasses all baselines in average accuracy (e.g., over 8.5% higher than the strong baseline Bespoke-Stratos-4B[†] that uses the same training data, and over 14% in other domains).

Table 4: Ablation study on the effect of conditioning the proposal distribution on y' .

Method	MATH500	AIME24	AIME25	AMC23	OlympiadBench	Avg
	Avg@2	Avg@32	Avg@32	Avg@32	Avg@2	Avg
Qwen3-4B-Base	45.30	4.79	5.73	27.73	23.37	21.38
Ours-4B	88.30	31.67	27.29	75.63	55.71	55.72
w/o y'	81.20	23.44	23.96	65.70	46.59	48.18

Method	GPQA-D	LCB-E	LCB-M	LCB-H	MMLU-Pro	Avg
	Avg@8	Avg@8	Avg@8	Avg@8	Avg@1	Avg
Qwen3-4B-Base	29.10	18.54	5.46	1.32	36.89	18.26
Ours-4B	45.33	80.29	33.68	5.79	65.53	46.12
w/o y'	40.53	67.93	16.63	2.44	61.49	37.80

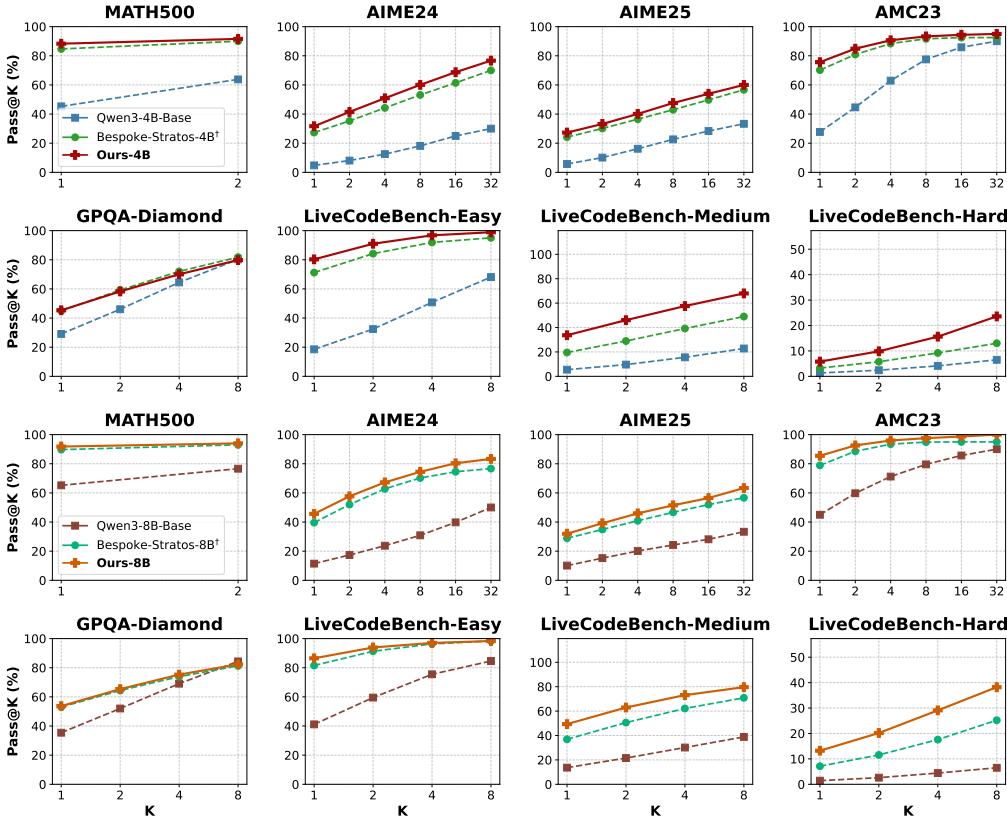


Figure 2: Pass@K comparison of baselines versus our method based on Qwen3-4B/8B-Base.

Notably, GPQA-Diamond and MMLU-Pro can be considered out-of-distribution test sets, as our training data only cover math and code, whereas they are in-domain for General-Reasoner-4B. Despite this, our method significantly outperforms General-Reasoner on these benchmarks, suggesting that the reasoning improvements from variational reasoning generalize effectively.

Additionally, our method demonstrates robustness across different prompt templates. Performance remains consistent between Prompt Template A and B (denoted as “-PA” and “-PB” in Table 6), with both outperforming baselines. Across four model scales, the accuracy-based estimator (“-Acc”) and the geometric mean of token likelihood estimator (“-GML”) exhibit similar performance, though the accuracy-based variant shows a slight advantage in math-related benchmarks.

Pass@K analysis. We report Pass@K results of experiments based on Qwen3-4B/8B-Base for different values of K. Figure 2 reveals two key trends: (1) Our method’s advantage increases with larger K on complex benchmarks (e.g., LiveCodeBench-Hard), and (2) Performance gaps diminish on simpler tasks (e.g., LiveCodeBench-Easy) and multiple-choice questions (e.g., GPQA-Diamond). This aligns with expectations, as simpler tasks offer limited room for improvement, and multiple-choice formats inherently allow high Pass@K with sufficiently large K. These results underscore the strong potential of variational reasoning in tackling complex tasks.

Table 5: Ablation study on effects of different $\pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}_k)$ estimators. Experiments are done in data 1k setting. Acc: accuracy; GML: geometric mean of token likelihood; L: naive likelihood.

Method	MATH500	AIME24	AIME25	AMC23	OlympiadBench	Avg
	Avg@2	Avg@32	Avg@32	Avg@32	Avg@2	Avg
Qwen2.5-7B-Instruct	75.60	10.94	7.40	51.10	39.91	36.99
Bespoke-Stratos-7B-1K [†]	77.20	16.25	13.96	53.75	40.88	40.41
Ours-Acc-7B-1K	81.30	19.69	18.44	<u>61.64</u>	45.99	45.41
Ours-GML-7B-1K	81.30	<u>19.27</u>	<u>18.33</u>	62.50	<u>45.48</u>	45.38
Ours-L-7B-1K	<u>79.90</u>	17.81	14.17	59.53	43.62	43.01

Method	GPQA-D	LCB-E	LCB-M	LCB-H	MMLU-Pro	Avg
	Avg@8	Avg@8	Avg@8	Avg@8	Avg@1	Avg
Qwen2.5-7B-Instruct	29.99	62.50	18.20	3.35	48.20	32.45
Bespoke-Stratos-7B-1K [†]	37.94	60.37	13.59	1.22	56.07	33.84
Ours-Acc-7B-1K	<u>41.16</u>	68.13	<u>21.42</u>	1.42	<u>60.94</u>	<u>38.62</u>
Ours-GML-7B-1K	41.35	68.41	23.30	2.74	61.31	39.42
Ours-L-7B-1K	39.90	66.42	19.90	1.93	58.61	37.35

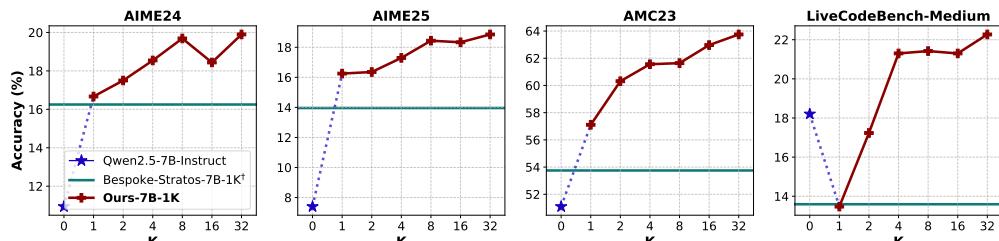


Figure 3: Effects of scaling up the number of thinking traces (K in Algorithm 1) sampled from variational posterior q_ϕ on the performance of the final reasoning model π_θ .

Training dynamics. We monitor training loss and gradient norms during training for Qwen3-4B/8B-base models (see Figure 1). Compared to Bespoke-Stratos-4B/8B[†], our method yields lower average training loss and fewer gradient norm spikes, indicating greater training stability. We attribute this stability to the π_θ/q_ϕ ratio in $\tilde{\rho}_k$. Specifically, for a reasoning trace \mathbf{z}_k , the weight $\tilde{\rho}_k$ is large when the trace is both high-quality (high $\pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}_k)$) and aligned with the reasoning policy (high likelihood ratio $\pi_\theta(\mathbf{z}_k|\mathbf{x})/q_\phi(\mathbf{z}_k|\mathbf{x}, \mathbf{y}')$). This adaptive weighting promotes stable and effective training.

4.3 ABLATION STUDIES

As mentioned in Section 4.1, we conduct ablation studies on both the 17k and 1k data settings to better analyze variational reasoning. Additional ablations are provided in Appendix F.2.

Scaling the number of thinking traces \mathbf{z}_k . We investigate the effect of increasing the number of traces \mathbf{z} sampled from the variational posterior q_ϕ (i.e., K in Algorithm 1) on the performance of the reasoning model π_θ . Experiments are conducted under the 1k data setting and faithful to Algorithm 1. We scale K exponentially from 1 to 32, adjusting the batch size to keep optimization steps consistent. Results in Figure 3 suggest that increasing K can further enhance model performance. This implies a practical trade-off between training computational cost and reasoning accuracy when selecting K .

Conditioning on \mathbf{y}' . We ablate the necessity of conditioning on \mathbf{y}' in data 17k setting. The variant w/o \mathbf{y}' samples thinking traces \mathbf{z} by the initial reasoning model instead of the variational posterior. Results (Table 4) show that removing \mathbf{y}' as the condition negatively affects the performance.

Different $\pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}_k)$ estimators. We ablate different estimators for $\pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}_k)$ used in the weight $\tilde{\rho}_k$ in data 1k setting (Table 5). We find estimators based on accuracy or geometric mean of token likelihood outperform the naive likelihood by a large margin, validating our analysis in Sec. 2.2.

5 CONCLUSION

We introduced a variational reasoning framework as a principled and stable objective for training reasoning models, while clarifying biases in existing SFT/RFT and RL methods. Beyond consistent gains over strong baselines on diverse reasoning tasks, our analysis offers a probabilistic perspective for interpreting current approaches. A natural future direction is extending training beyond a single round ($T > 1$ in Algorithm 1) and exploring richer posterior design for the answer hint \mathbf{y}' .

486 REFERENCES
487

488 Yoshua Bengio, Salem Lahou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio.
489 Gflownet foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023. URL <http://jmlr.org/papers/v24/22-0364.html>.
490

491 Jörg Bornschein and Yoshua Bengio. Reweighted wake-sleep. In *International Conference on
492 Learning Representations (ICLR)*, 2015.
493

494 Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv
495 preprint arXiv:1509.00519*, 2015.
496

497 Edoardo Cetin, Tianyu Zhao, and Yujin Tang. Reinforcement learning teachers of test time scaling.
498 *arXiv preprint arXiv:2506.08388*, 2025.
499

500 Haolin Chen, Yihao Feng, Zuxin Liu, Weiran Yao, Akshara Prabhakar, Shelby Heinecke, Ricky
501 Ho, Phil Mui, Silvio Savarese, Caiming Xiong, et al. Language models are hidden reasoners:
502 Unlocking latent reasoning capabilities via self-rewarding. *arXiv preprint arXiv:2411.04282*, 2024.
503

504 Yang Chen, Zhuolin Yang, Zihan Liu, Chankyu Lee, Peng Xu, Mohammad Shoeybi, Bryan Catanzaro,
505 and Wei Ping. Acereason-nemotron: Advancing math and code reasoning through reinforcement
506 learning. *arXiv preprint arXiv:2505.16400*, 2025.
507

508 Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu
509 Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025.
510

511 Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V
512 Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation
513 model post-training. *arXiv preprint arXiv:2501.17161*, 2025.
514

515 Gheorghe Comanici, Eric Bieber, Mike Schaeckermann, Ice Pasupat, Noveen Sachdeva, Inderjit
516 Dhillon, Marcel Blstein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier
517 with advanced reasoning, multimodality, long context, and next generation agentic capabilities.
518 *arXiv preprint arXiv:2507.06261*, 2025.
519

520 Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu
521 Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint
522 arXiv:2502.01456*, 2025a.
523

524 Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen
525 Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for
526 reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025b.
527

528 Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao,
529 Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative
530 foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.
531

532 Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weilin Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang,
533 and Yi Wu. On designing effective rl reward at training time for llm reasoning. *arXiv preprint
534 arXiv:2410.15115*, 2024.
535

536 Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna
537 Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning
538 models. *arXiv preprint arXiv:2506.04178*, 2025.
539

540 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
541 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms
542 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
543

544 Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu,
545 Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiad-
546 bench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal
547 scientific problems, 2024.
548

540 Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin
 541 Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring coding chal-
 542 lenge competence with APPS. In *Thirty-fifth Conference on Neural Information Processing*
 543 *Systems Datasets and Benchmarks Track (Round 2)*, 2021a. URL <https://openreview.net/forum?id=sD93GOzH3i5>.

544

545 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,
 546 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*,
 547 2021b.

548

549 Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Uddandarao, Samuel Albanie, Ameya Prabhu, and
 550 Matthias Bethge. A sober look at progress in language model reasoning: Pitfalls and paths to
 551 reproducibility. *arXiv preprint arXiv:2504.07086*, 2025.

552

553 Matthew Douglas Hoffman, Du Phan, david dohan, Sholto Douglas, Tuan Anh Le, Aaron T Parisi,
 554 Pavel Sountsov, Charles Sutton, Sharad Vikram, and Rif A. Saurous. Training chain-of-thought via
 555 latent-variable inference. In *Thirty-seventh Conference on Neural Information Processing Systems*,
 556 2023. URL <https://openreview.net/forum?id=a147pIS2Co>.

557

558 Edward J Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio,
 559 and Nikolay Malkin. Amortizing intractable inference in large language models. In *The Twelfth*
 560 *International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=Ouj6p4ca60>.

561

562 Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv*
 563 *preprint arXiv:2501.03262*, 2025.

564

565 Jingcheng Hu, Yinmin Zhang, Qi Han, Dixin Jiang, and Heung-Yeung Shum Xiangyu Zhang. Open-
 566 reasoner-zero: An open source approach to scaling reinforcement learning on the base model.
 567 <https://github.com/Open-Reasoner-Zero/Open-Reasoner-Zero>, 2025.

568

569 Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec
 570 Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint*
 571 *arXiv:2412.16720*, 2024.

572

573 Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando
 574 Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free eval-
 575 uation of large language models for code. In *The Thirteenth International Conference on Learning*
 576 *Representations*, 2025. URL <https://openreview.net/forum?id=chfJJYC3iL>.

577

578 Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai
 579 Zou, Chao Du, Tianyu Pang, and Wenhui Chen. Verltool: Towards holistic agentic reinforcement
 580 learning with tool use. *arXiv preprint arXiv:2509.01055*, 2025.

581

582 Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and
 583 Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement
 584 learning. *arXiv preprint arXiv:2503.09516*, 2025.

585

586 Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy,
 587 Aaron Courville, and Nicolas Le Roux. Vineppo: Unlocking rl potential for llm reasoning through
 588 refined credit assignment. *arXiv preprint arXiv:2410.01679*, 2024.

589

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

592

593 Woosuk Kwon, Zuhuan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E.
 594 Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model
 595 serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating*
 596 *Systems Principles*, 2023.

597

Bespoke Labs. Bespoke-stratos: The unreasonable effectiveness of reasoning distilla-
 598 tion. [www.bespokelabs.ai/blog/bespoke-stratos-the-unreasonable-effectiveness-of-reasoning-
 599 distillation](http://www.bespokelabs.ai/blog/bespoke-stratos-the-unreasonable-effectiveness-of-reasoning-distillation), 2025. Accessed: 2025-01-22.

594 Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review.
 595 *arXiv preprint arXiv:1805.00909*, 2018.

596

597 Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Eric Tang, Sumanth Hegde, Kourosh
 598 Hakhamaneshi, Shishir G Patil, Matei Zaharia, et al. Llms can easily learn to reason from
 599 demonstrations structure, not content, is what matters! *arXiv preprint arXiv:2502.07374*, 2025a.

600 Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif
 601 Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in
 602 ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*,
 603 13:9, 2024.

604 Junlong Li, Daya Guo, Dejian Yang, Runxin Xu, Yu Wu, and Junxian He. Codei/o: Condensing
 605 reasoning patterns via code input-output prediction. *arXiv preprint arXiv:2502.07316*, 2025b.

606

607 Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and
 608 Ge Li. Taco: Topics in algorithmic code generation dataset. *arXiv preprint arXiv:2312.14852*,
 609 2023.

610 Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan
 611 Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth
 612 International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=v8L0pN6EOi>.

613

614 Jiawei Liu and Lingming Zhang. Code-r1: Reproducing r1 for code with reliable rewards. <https://github.com/ganler/code-r1>, 2025.

615

616 Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min
 617 Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*,
 618 2025.

619

620 Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference
 621 on Learning Representations*, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.

622

623 Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai,
 624 Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview
 625 with a 1.5b model by scaling rl, 2025. Notion Blog.

626

627 Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhui Chen. General-reasoner:
 628 Advancing llm reasoning across all domains. *arXiv preprint arXiv:2505.14652*, 2025.

629

630 Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman,
 631 Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding
 632 large language models. *arXiv preprint arXiv:2310.12931*, 2023.

633

634 MAA. American mathematics competitions - amc. <https://maa.org/>, 2023.

635

636 MAA. American invitational mathematics examination - aime. <https://maa.org/>, 2025.

637

638 Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke
 639 Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time
 scaling. *arXiv preprint arXiv:2501.19393*, 2025.

640

641 Penghui Qi, Zichen Liu, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Optimizing anytime
 reasoning via budget relative policy optimization. *arXiv preprint arXiv:2505.13438*, 2025.

642

643 Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea
 644 Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances
 645 in Neural Information Processing Systems*, 36:53728–53741, 2023.

646

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

648 Yangjun Ruan, Neil Band, Chris J Maddison, and Tatsunori Hashimoto. Reasoning to learn from
 649 latent thoughts. *arXiv preprint arXiv:2503.18866*, 2025.
 650

651 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
 652 Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical
 653 reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

654 Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's razor: Why online reinforcement learning
 655 forgets less. *arXiv preprint arXiv:2509.04259*, 2025.
 656

657 Yunhao Tang, Sid Wang, and Rémi Munos. Learning to chain-of-thought with jensen's evidence
 658 lower bound. *arXiv preprint arXiv:2503.19618*, 2025.

659 Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun
 660 Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with
 661 llms. *arXiv preprint arXiv:2501.12599*, 2025.

662 NovaSky Team. Sky-t1: Train your own o1 preview model within \$450. <https://novasky-ai.github.io/posts/sky-t1>, 2025a. Accessed: 2025-01-09.
 663

664 Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown. *Hugging Face*, 2024.
 665

666 Qwen Team. Qwen3, April 2025b. URL <https://qwenlm.github.io/blog/qwen3/>.
 667

668 Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay
 669 Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation
 670 and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
 671

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

675 Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming
 676 Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-
 677 task language understanding benchmark. In *The Thirty-eight Conference on Neural Information
 678 Processing Systems Datasets and Benchmarks Track*, 2024.

679 Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin,
 680 Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. Ragen: Understanding self-evolution in llm
 681 agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025.
 682

683 Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried,
 684 Gabriel Synnaeve, Rishabh Singh, and Sida I Wang. Swe-rl: Advancing llm reasoning via
 685 reinforcement learning on open software evolution. *arXiv preprint arXiv:2502.18449*, 2025.

686 Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu
 687 Tang, Xiaowei Lv, et al. Light-rl: Curriculum sft, dpo and rl for long cot from scratch and beyond.
 688 *arXiv preprint arXiv:2503.10460*, 2025.

689 Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu,
 690 Lu Qi, Ming-Hsuan Yang, and Xu Yang. On the generalization of sft: A reinforcement learning
 691 perspective with reward rectification. *arXiv preprint arXiv:2508.05629*, 2025.
 692

693 Violet Xiang, Charlie Snell, Kanishk Gandhi, Alon Albalak, Anikait Singh, Chase Blagden, Duy
 694 Phung, Rafael Rafailov, Nathan Lile, Dakota Mahan, et al. Towards system 2 reasoning in llms:
 695 Learning how to think with meta chain-of-thought. *arXiv preprint arXiv:2501.04682*, 2025.

696 Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu,
 697 Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement
 698 learning. *arXiv preprint arXiv:2502.14768*, 2025.
 699

700 Yifei Xu, Tusher Chakraborty, Srinagesh Sharma, Leonardo Nunes, Emre Kıcıman, Songwu Lu, and
 701 Ranveer Chandra. Direct reasoning optimization: Llms can reward and refine their own reasoning
 for open-ended tasks. *arXiv preprint arXiv:2506.13351*, 2025.

702 Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. Sim-
 703 pletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. *arXiv preprint*
 704 *arXiv:2509.02479*, 2025.

705 An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li,
 706 Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin
 707 Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang,
 708 Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia,
 709 Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu
 710 Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*,
 711 2024.

712 John Yang, Kilian Lieret, Carlos E Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang,
 713 Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. Swe-smith: Scaling data for software
 714 engineering agents. *arXiv preprint arXiv:2504.21798*, 2025.

715 Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong
 716 Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale.
 717 *arXiv preprint arXiv:2503.14476*, 2025a.

718 Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao,
 719 Zhiyuan Liu, et al. Rlpr: Extrapolating rlvr to general domains without verifiers. *arXiv preprint*
 720 *arXiv:2506.18254*, 2025b.

721 Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does
 722 reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv*
 723 *preprint arXiv:2504.13837*, 2025a.

724 Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, et al. Vapo: Efficient and
 725 reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*,
 726 2025b.

727 Weihao Zeng, Yuzhen Huang, Wei Liu, Keqing He, Qian Liu, Zejun Ma, and Junxian He. 7b model
 728 and 8k examples: Emerging reasoning with reinforcement learning is both effective and efficient.
 729 <https://hkust-nlp.notion.site/simplerl-reason>, 2025. Notion Blog.

730 Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang,
 731 Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint*
 732 *arXiv:2507.18071*, 2025.

733 Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyuan Luo, Zhangchi Feng, and
 734 Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Pro-
 735 ceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3:
 736 System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics.
 737 URL <http://arxiv.org/abs/2403.13372>.

738 Han Zhong, Yutong Yin, Shenao Zhang, Xiaojun Xu, Yuanxin Liu, Yifei Zuo, Zhihan Liu, Boyi
 739 Liu, Sirui Zheng, Hongyi Guo, et al. Brite: Bootstrapping reinforced thinking process to enhance
 740 language model reasoning. *arXiv preprint arXiv:2501.18858*, 2025.

741 Xiangxin Zhou, Zichen Liu, Anya Sims, Haonan Wang, Tianyu Pang, Chongxuan Li, Liang
 742 Wang, Min Lin, and Chao Du. Reinforcing general reasoning without verifiers. *arXiv preprint*
 743 *arXiv:2505.21493*, 2025.

744

745

746

747

748

749

750

751

752

753

754

755

756 A DETAILED DERIVATIONS
757758 In this section, we provide detailed derivations of the conclusions presented in the main text, along
759 with some additional results.
760761 A.1 DERIVATION FOR EQ. (4)
762763 The ELBO objective induced from Eq. (2) can be rewritten as
764

765
$$\begin{aligned} \mathcal{L}_{\text{ELBO}}(\mathbf{x}, \mathcal{Y}_{\mathbf{x}}, \mathbf{y}'; \pi_{\theta}, q_{\phi}) &\triangleq \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')} [\log \pi_{\theta}(\mathcal{Y}_{\mathbf{x}}|\mathbf{x}, \mathbf{z})] - \text{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')||\pi_{\theta}(\mathbf{z}|\mathbf{x})) \\ &= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')} [\log \pi_{\theta}(\mathbf{z}, \mathcal{Y}_{\mathbf{x}}|\mathbf{x})] + \mathcal{H}(q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')) \\ &= \log P_{\theta}(\mathcal{Y}_{\mathbf{x}}|\mathbf{x}) - \mathbb{D}_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')||P_{\theta}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_{\mathbf{x}})), \end{aligned} \quad (13)$$

766

767 where $\mathcal{H}(\cdot)$ is entropy function and $P_{\theta}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_{\mathbf{x}}) = \frac{\pi_{\theta}(\mathcal{Y}_{\mathbf{x}}|\mathbf{x}, \mathbf{z})\pi_{\theta}(\mathbf{z}|\mathbf{x})}{P_{\theta}(\mathcal{Y}_{\mathbf{x}}|\mathbf{x})}$ is the true posterior distribution.
768769 A.2 DERIVATION FOR EQ. (6)
770771 Given the IWAE-style lower bound $\mathcal{L}_{\text{ELBO}}^K(\mathbf{x}, \mathcal{Y}_{\mathbf{x}}, \mathbf{y}'; \pi_{\theta}, q_{\phi})$ in Eq. (5), we can derive its gradient
772 w.r.t. model parameters θ as:
773

774
$$\begin{aligned} &\nabla_{\theta} \mathcal{L}_{\text{ELBO}}^K(\mathbf{x}, \mathcal{Y}_{\mathbf{x}}, \mathbf{y}'; \pi_{\theta}, q_{\phi}) \\ &= \nabla_{\theta} \mathbb{E}_{\mathbf{z}_{1:K} \sim q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')} \left[\log \frac{1}{K} \sum_{k=1}^K \frac{\pi_{\theta}(\mathbf{z}_k, \mathcal{Y}_{\mathbf{x}}|\mathbf{x})}{q_{\phi}(\mathbf{z}_k|\mathbf{x}, \mathbf{y}')} \right] \\ &= \mathbb{E}_{\mathbf{z}_{1:K} \sim q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')} \left[\frac{\frac{1}{K} \sum_{k=1}^K \frac{\nabla_{\theta} \pi_{\theta}(\mathbf{z}_k, \mathcal{Y}_{\mathbf{x}}|\mathbf{x})}{q_{\phi}(\mathbf{z}_k|\mathbf{x}, \mathbf{y}')}}{\frac{1}{K} \sum_{k=1}^K \frac{\pi_{\theta}(\mathbf{z}_k, \mathcal{Y}_{\mathbf{x}}|\mathbf{x})}{q_{\phi}(\mathbf{z}_k|\mathbf{x}, \mathbf{y}')}} \right] \\ &= \mathbb{E}_{\mathbf{z}_{1:K} \sim q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')} \left[\sum_{k=1}^K \tilde{\rho}_k \nabla_{\theta} \log \pi_{\theta}(\mathbf{z}_k, \mathcal{Y}_{\mathbf{x}}|\mathbf{x}) \right] \\ &\text{where } \tilde{\rho}_k = \frac{\rho_k}{\sum_{j=1}^K \rho_j} \quad \text{and} \quad \rho_k = \frac{\pi_{\theta}(\mathbf{z}_k, \mathcal{Y}_{\mathbf{x}}|\mathbf{x})}{q_{\phi}(\mathbf{z}_k|\mathbf{x}, \mathbf{y}')}. \end{aligned} \quad (14)$$

775

776 Using the notations of ρ_k and $\tilde{\rho}_k$, we can further derive the gradient w.r.t. ϕ as:
777

778
$$\begin{aligned} &\nabla_{\phi} \mathcal{L}_{\text{ELBO}}^K(\mathbf{x}, \mathcal{Y}_{\mathbf{x}}, \mathbf{y}'; \pi_{\theta}, q_{\phi}) \\ &= \nabla_{\phi} \mathbb{E}_{\mathbf{z}_{1:K} \sim q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')} \left[\log \frac{1}{K} \sum_{k=1}^K \frac{\pi_{\theta}(\mathbf{z}_k, \mathcal{Y}_{\mathbf{x}}|\mathbf{x})}{q_{\phi}(\mathbf{z}_k|\mathbf{x}, \mathbf{y}')} \right] \\ &= \mathbb{E}_{\mathbf{z}_{1:K} \sim q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')} \left[\left(\log \frac{1}{K} \sum_{k=1}^K \rho_k \right) \cdot \sum_{k=1}^K \nabla_{\phi} \log q_{\phi}(\mathbf{z}_k|\mathbf{x}, \mathbf{y}') - \sum_{k=1}^K \tilde{\rho}_k \nabla_{\phi} \log q_{\phi}(\mathbf{z}_k|\mathbf{x}, \mathbf{y}') \right] \\ &= \mathbb{E}_{\mathbf{z}_{1:K} \sim q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}')} \left[\sum_{k=1}^K \left(-\tilde{\rho}_k + \log \frac{1}{K} \sum_{k=1}^K \rho_k \right) \cdot \nabla_{\phi} \log q_{\phi}(\mathbf{z}_k|\mathbf{x}, \mathbf{y}') \right]. \end{aligned}$$

779

800 A.3 PROOF OF THEOREM 1
801802 As to the computation of $\pi_{\theta}(\mathcal{Y}_{\mathbf{x}}|\mathbf{x}, \mathbf{z})$, there are two unbiased estimators:
803

804
$$\begin{aligned} \text{Likelihood-based estimator: } \pi_{\theta}(\mathcal{Y}_{\mathbf{x}}|\mathbf{x}, \mathbf{z}) &= |\mathcal{Y}_{\mathbf{x}}| \cdot \mathbb{E}_{\mathbf{y} \sim \mathcal{U}(\mathcal{Y}_{\mathbf{x}})} [\pi_{\theta}(\mathbf{y}|\mathbf{x}, \mathbf{z})]; \\ \text{Accuracy-based estimator: } \pi_{\theta}(\mathcal{Y}_{\mathbf{x}}|\mathbf{x}, \mathbf{z}) &= \mathbb{E}_{\mathbf{y} \sim \pi_{\theta}(\mathbf{y}|\mathbf{x}, \mathbf{z})} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_{\mathbf{x}})], \end{aligned} \quad (15)$$

805

806 where $|\mathcal{Y}_{\mathbf{x}}|$ is the cardinal (number of elements) of $\mathcal{Y}_{\mathbf{x}}$, $\mathcal{U}(\mathcal{Y}_{\mathbf{x}})$ is the uniform distribution on $\mathcal{Y}_{\mathbf{x}}$,
807 and $\mathbb{1}(\cdot)$ is the indicator function. When $|\mathcal{Y}_{\mathbf{x}}| = 1$, i.e., there is a unique correct answer expression
808

\mathbf{y}^* , Zhou et al. (2025) show that the likelihood-based estimator has lower variance (in fact, zero) compared to the accuracy-based one. We now extend this comparison to general cases when $|\mathcal{Y}_x| > 1$:

$$\begin{aligned} \text{Var}_{\text{like}} &= |\mathcal{Y}_x|^2 \cdot \text{Var}_{\mathbf{y} \sim \mathcal{U}(\mathcal{Y}_x)} [\pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})]; \\ \text{Var}_{\text{acc}} &= \pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}) \cdot (1 - \pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z})). \end{aligned} \quad (16)$$

Note that the variance Var_{acc} of accuracy-based estimator is independent of $|\mathcal{Y}_x|$ and the model distribution π_θ over different elements in \mathcal{Y}_x . Assuming that in the worst case where only one element $\mathbf{y}^* \in \mathcal{Y}_x$ has non-zero probability under π_θ , i.e., $\pi_\theta(\mathbf{y}^*|\mathbf{x}, \mathbf{z}) = \pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z})$, we have

$$\begin{aligned} \text{Var}_{\text{like}}^{\text{worst}} &\triangleq \max_{\pi_\theta} \text{Var}_{\text{like}} = (|\mathcal{Y}_x| - 1) \cdot \pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z})^2; \\ \text{Var}_{\text{acc}}^{\text{worst}} &\triangleq \max_{\pi_\theta} \text{Var}_{\text{acc}} = \pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}) \cdot (1 - \pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z})). \end{aligned} \quad (17)$$

Here we slightly abuse the notation of \max_{π_θ} , since the maximization is taken w.r.t. all π_θ under fixed value $\pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z})$. As seen, $\text{Var}_{\text{like}}^{\text{worst}} \geq \text{Var}_{\text{acc}}^{\text{worst}}$ holds when the model accuracy (condition on \mathbf{x}, \mathbf{z}) satisfies

$$\pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}) \geq \frac{1}{|\mathcal{Y}_x|}, \quad (18)$$

which almost always holds for $|\mathcal{Y}_x| \gg 1$. \square

A.4 DERIVATION FOR EQ. (9)

Now we derive the gradient of the forward KL divergence $\mathbb{D}_{\text{KL}}(P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_x) \parallel q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}'))$ w.r.t. ϕ :

$$\begin{aligned} &\nabla_\phi \mathbb{D}_{\text{KL}}(P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_x) \parallel q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')) \\ &= -\nabla_\phi \mathbb{E}_{P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_x)} [\log q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')] \\ &= -\nabla_\phi \mathbb{E}_{\pi_\theta(\mathbf{z}|\mathbf{x})} \left[\frac{\pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z})}{P_\theta(\mathcal{Y}_x|\mathbf{x})} \log q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}') \right] \\ &\simeq \mathbb{E}_{\mathbf{z}_{1:M} \sim \pi_\theta(\mathbf{z}|\mathbf{x})} \left[\sum_{m=1}^M \tilde{w}_m \nabla_\phi \log q_\phi(\mathbf{z}_m|\mathbf{x}, \mathbf{y}') \right] \triangleq \nabla_\phi \mathcal{L}_{\text{forward}}^M, \\ &\text{where } \tilde{w}_m = \frac{w_m}{\sum_{j=1}^M w_j} \quad \text{and} \quad w_m = \pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}_m). \end{aligned} \quad (19)$$

A.5 CONNECTION TO MORE GENERAL RL REWARD SHAPING

In the literature on reinforcement learning with verifiable rewards (RLVR), various strategies for reward shaping have been proposed, many of which can be expressed as

$$\mathcal{R}(\mathbf{x}, \mathbf{y}) = \begin{cases} \alpha & \text{if } \mathbf{y} \in \mathcal{Y}_x; \\ \beta & \text{if } \mathbf{y} \notin \mathcal{Y}_x \wedge \mathbf{y} \in \mathcal{Y}_{\text{format}}; \\ \gamma & \text{otherwise,} \end{cases} \quad (20)$$

where α, β, γ are hyperparameters, $\mathcal{Y}_{\text{format}}$ is the set of answers that correctly follow required format (e.g., \boxed{\ }) and is typically independent of \mathbf{x} . Apparently, $\mathcal{Y}_x \subset \mathcal{Y}_{\text{format}}$ holds for any \mathbf{x} . Then we can derive the gradient of training objective under the general reward shaping $\mathcal{R}(\mathbf{x}, \mathbf{y})$ as

$$\begin{aligned} &\nabla_\theta \mathcal{L}_{\text{general-RL}}(\mathbf{x}, \pi_\theta) \\ &\triangleq \nabla_\theta \mathbb{E}_{\pi_\theta(\mathbf{z}|\mathbf{x})} \mathbb{E}_{\pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})} [\mathcal{R}(\mathbf{x}, \mathbf{y})] \\ &= \mathbb{E}_{\pi_\theta(\mathbf{z}|\mathbf{x})} \mathbb{E}_{\pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})} [\mathcal{R}(\mathbf{x}, \mathbf{y}) \cdot (\nabla_\theta \log \pi_\theta(\mathbf{z}|\mathbf{x}) + \nabla_\theta \log \pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}))] \\ &\xrightarrow{\text{only w.r.t. } \pi_\theta(\mathbf{z}|\mathbf{x})} \mathbb{E}_{\pi_\theta(\mathbf{z}|\mathbf{x})} \mathbb{E}_{\pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})} [\mathcal{R}(\mathbf{x}, \mathbf{y}) \cdot \nabla_\theta \log \pi_\theta(\mathbf{z}|\mathbf{x})] \\ &= \mathbb{E}_{\pi_\theta(\mathbf{z}|\mathbf{x})} [((\alpha - \beta) \cdot \pi_\theta(\mathcal{Y}_x|\mathbf{x}, \mathbf{z}) + (\beta - \gamma) \cdot \pi_\theta(\mathcal{Y}_{\text{format}}|\mathbf{x}, \mathbf{z})) \cdot \nabla_\theta \log \pi_\theta(\mathbf{z}|\mathbf{x})] \\ &= (\alpha - \beta) \cdot P_\theta(\mathcal{Y}_x|\mathbf{x}) \cdot \nabla_\theta \mathbb{D}_{\text{KL}}(P_\theta^{\text{sg}}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_x) \parallel \pi_\theta(\mathbf{z}|\mathbf{x})) \\ &\quad + (\beta - \gamma) \cdot P_\theta(\mathcal{Y}_{\text{format}}|\mathbf{x}) \cdot \nabla_\theta \mathbb{D}_{\text{KL}}(P_\theta^{\text{sg}}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_{\text{format}}) \parallel \pi_\theta(\mathbf{z}|\mathbf{x})), \end{aligned} \quad (21)$$

864 where $P_\theta(\mathcal{Y}_{\text{format}}|\mathbf{x})$ is the probability that the output answers follow the required format. It is easy to
 865 know that the optimal solution for Eq. (21) can be written as:
 866

$$\pi_\theta^*(\mathbf{z}|\mathbf{x}) = \frac{(\alpha - \beta) \cdot P_\theta^{\text{sg}}(\mathbf{z}, \mathcal{Y}_{\mathbf{x}}|\mathbf{x}) + (\beta - \gamma) \cdot P_\theta^{\text{sg}}(\mathbf{z}, \mathcal{Y}_{\text{format}}|\mathbf{x})}{(\alpha - \beta) \cdot P_\theta^{\text{sg}}(\mathcal{Y}_{\mathbf{x}}|\mathbf{x}) + (\beta - \gamma) \cdot P_\theta^{\text{sg}}(\mathcal{Y}_{\text{format}}|\mathbf{x})}. \quad (22)$$

869 **Remark.** When $\beta = \gamma$, i.e., there is no format reward, the optimization problem in Eq. (21) degrades
 870 to Eq. (11). When $\alpha > \beta > \gamma$, the model $\pi_\theta(\mathbf{z}|\mathbf{x})$ will tend to hack reward function on hard problems
 871 (i.e., low $P_\theta(\mathcal{Y}_{\mathbf{x}}|\mathbf{x})$) that can easily follow format (i.e., high $P_\theta(\mathcal{Y}_{\text{format}}|\mathbf{x})$), where $\pi_\theta(\mathbf{z}|\mathbf{x})$ will seek
 872 modes of $P_\theta^{\text{sg}}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_{\text{format}})$. Besides, there may be an intuition that setting $\beta < 0$ could alleviate
 873 reward hacking, however, as shown in Eq. (21), the optimization only depends on the relative values
 874 of $\alpha - \beta$ and $\beta - \gamma$.

875 Now we show that it is straightforward to debias $P_\theta(\mathcal{Y}_{\mathbf{x}}|\mathbf{x})$ and $P_\theta(\mathcal{Y}_{\text{format}}|\mathbf{x})$ in Eq. (21). Specifically,
 876 we can rewrite the reward function as (note that reward functions are equivalent up to any constant):
 877

$$\mathcal{R}(\mathbf{x}, \mathbf{y}) = (\alpha - \beta) \cdot \mathbb{1}(\mathbf{y} \in \mathcal{Y}_{\mathbf{x}}) + (\beta - \gamma) \cdot \mathbb{1}(\mathbf{y} \in \mathcal{Y}_{\text{format}}). \quad (23)$$

879 Then the debiased version of reward function is
 880

$$\mathcal{R}^\dagger(\mathbf{x}, \mathbf{y}) = \frac{(\alpha - \beta)}{P_\theta^{\text{sg}}(\mathcal{Y}_{\mathbf{x}}|\mathbf{x})} \cdot \mathbb{1}(\mathbf{y} \in \mathcal{Y}_{\mathbf{x}}) + \frac{(\beta - \gamma)}{P_\theta^{\text{sg}}(\mathcal{Y}_{\text{format}}|\mathbf{x})} \cdot \mathbb{1}(\mathbf{y} \in \mathcal{Y}_{\text{format}}), \quad (24)$$

883 where in practice $P_\theta^{\text{sg}}(\mathcal{Y}_{\mathbf{x}}|\mathbf{x})$ and $P_\theta^{\text{sg}}(\mathcal{Y}_{\text{format}}|\mathbf{x})$ can be approximated by the ratio of correct answers
 884 (i.e., model accuracy) and the ratio of correct format for each batch of RL rollouts (larger rollout
 885 number could lead to more accurate estimation). After using the debiased reward function $\mathcal{R}^\dagger(\mathbf{x}, \mathbf{y})$,
 886 the optimal solution of $\pi_\theta^*(\mathbf{z}|\mathbf{x})$ becomes

$$\pi_\theta^*(\mathbf{z}|\mathbf{x}) = \frac{(\alpha - \beta) \cdot P_\theta^{\text{sg}}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_{\mathbf{x}}) + (\beta - \gamma) \cdot P_\theta^{\text{sg}}(\mathbf{z}|\mathbf{x}, \mathcal{Y}_{\text{format}})}{\alpha - \gamma}. \quad (25)$$

890 A.6 SPECIAL CASES IN EQ. (6)

891 **Special case I:** $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}') = \pi_\theta^{\text{sg}}(\mathbf{z}|\mathbf{x})$. In this case, we can simplify the gradient estimation as
 892

$$\nabla_\theta \mathcal{L}_{\text{ELBO}}^K(\mathbf{x}, \mathcal{Y}_{\mathbf{x}}; \pi_\theta, \pi_\theta^{\text{sg}}) = \mathbb{E}_{\mathbf{z}_{1:K} \sim \pi_\theta(\mathbf{z}|\mathbf{x})} \left[\sum_{k=1}^K \tilde{w}_k \nabla_\theta \log \pi_\theta(\mathbf{z}_k, \mathcal{Y}_{\mathbf{x}}|\mathbf{x}) \right] \quad (26)$$

893 where $\tilde{w}_k = \frac{w_k}{\sum_{j=1}^K w_j}$ and $w_k = \pi_\theta(\mathcal{Y}_{\mathbf{x}}|\mathbf{x}, \mathbf{z}_k)$,
 894

895 which can be regarded as a normalized version of VeriFree (Zhou et al., 2025).

896 **Special case II:** $K = 1$. In this case $\mathcal{L}_{\text{ELBO}}^1 = \mathcal{L}_{\text{ELBO}}$ and we can simplify the gradient estimation as
 897

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\text{ELBO}}(\mathbf{x}, \mathcal{Y}_{\mathbf{x}}, \mathbf{y}'; \pi_\theta, q_\phi) &= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')} [\nabla_\theta \log \pi_\theta(\mathbf{z}, \mathcal{Y}_{\mathbf{x}}|\mathbf{x})]; \\ \nabla_\phi \mathcal{L}_{\text{ELBO}}(\mathbf{x}, \mathcal{Y}_{\mathbf{x}}, \mathbf{y}'; \pi_\theta, q_\phi) &= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')} \left[\left(\log \frac{\pi_\theta(\mathbf{z}, \mathcal{Y}_{\mathbf{x}}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')} \right) \cdot \nabla_\phi \log q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}') \right]. \end{aligned} \quad (27)$$

906 A.7 MORE DERIVATIONS FOR EQ. (11)

907 Now we investigate the gradient of binary reward RL w.r.t. $\pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})$:

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\text{bi-RL}}(\mathbf{x}, \pi_\theta) &\triangleq \nabla_\theta \mathbb{E}_{\pi_\theta(\mathbf{z}|\mathbf{x})} \mathbb{E}_{\pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_{\mathbf{x}})] \\ &= \mathbb{E}_{\pi_\theta(\mathbf{z}|\mathbf{x})} \mathbb{E}_{\pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_{\mathbf{x}}) \cdot (\nabla_\theta \log \pi_\theta(\mathbf{z}|\mathbf{x}) + \nabla_\theta \log \pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}))] \\ &\xrightarrow[\pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})]{\text{only w.r.t.}} \mathbb{E}_{\pi_\theta(\mathbf{z}|\mathbf{x})} \mathbb{E}_{\pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_{\mathbf{x}}) \cdot \nabla_\theta \log \pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})] \\ &= \mathbb{E}_{\pi_\theta(\mathbf{z}|\mathbf{x})} [\nabla_\theta \pi_\theta(\mathcal{Y}_{\mathbf{x}}|\mathbf{x}, \mathbf{z})], \end{aligned} \quad (28)$$

916 where the optimal solution is straightforward that $\forall \mathbf{z}$, there is $\pi_\theta^*(\mathcal{Y}_{\mathbf{x}}|\mathbf{x}, \mathbf{z}) = 1$. However, this
 917 optimal solution is usually unachievable, since it requires the model to return 100% correct answers
 918 independent of the thinking process \mathbf{z} .

918 A.8 APPLYING VARIATIONAL POSTERIOR FOR RL
919920 Recall that the objective function of binary reward RL is defined as
921

922
$$\begin{aligned} \mathcal{L}_{\text{bi-RL}}(\mathbf{x}, \pi_\theta) &\triangleq \mathbb{E}_{\pi_\theta(\mathbf{z}|\mathbf{x})} \mathbb{E}_{\pi_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})} [\mathbb{1}(\mathbf{y} \in \mathcal{Y}_\mathbf{x})] \\ 923 &= \mathbb{E}_{\pi_\theta(\mathbf{z}|\mathbf{x})} [\pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z})]. \end{aligned} \quad (29)$$

924 Suppose the data points are drawn from a behavior policy $q(\mathbf{z}|\mathbf{x})$, the RL objective can be reformulated
925 using an importance sampling correction term as follows:
926

927
$$\mathcal{L}_{\text{bi-RL}}(\mathbf{x}, \pi_\theta) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[\frac{\pi_\theta(\mathbf{z}|\mathbf{x})}{q(\mathbf{z}|\mathbf{x})} \pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}) \right]. \quad (30)$$

928

929 Then, a natural question arises: *what is the optimal behavior policy $q(\mathbf{z}|\mathbf{x})$ that minimizes the
930 variance of estimating $\mathcal{L}_{\text{bi-RL}}(\mathbf{x}, \pi_\theta)$?* Specifically, we can compute
931

932
$$\begin{aligned} \text{Var}_{q(\mathbf{z}|\mathbf{x})} \left[\frac{\pi_\theta(\mathbf{z}|\mathbf{x})}{q(\mathbf{z}|\mathbf{x})} \pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}) \right] &= \text{Var}_{q(\mathbf{z}|\mathbf{x})} \left[\frac{P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x})}{q(\mathbf{z}|\mathbf{x})} P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}) \right] \\ 933 &= P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x})^2 \cdot \text{Var}_{q(\mathbf{z}|\mathbf{x})} \left[\frac{P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x})}{q(\mathbf{z}|\mathbf{x})} \right] \\ 934 &= P_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x})^2 \cdot \left(\sum_{\mathbf{z}} \frac{P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x})^2}{q(\mathbf{z}|\mathbf{x})} - 1 \right), \end{aligned} \quad (31)$$

935
936
937
938

939 which is equivalent to minimizing
940

941
$$\min_{q(\mathbf{z}|\mathbf{x})} \sum_{\mathbf{z}} \frac{P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x})^2}{q(\mathbf{z}|\mathbf{x})} \quad \text{s.t.} \quad \sum_{\mathbf{z}} q(\mathbf{z}|\mathbf{x}) = 1, q(\mathbf{z}|\mathbf{x}) \geq 0. \quad (32)$$

942

943 Using calculus of variations with a Lagrange multiplier λ , we obtain
944

945
$$\begin{aligned} \delta \left[\sum_{\mathbf{z}} \frac{P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x})^2}{q(\mathbf{z}|\mathbf{x})} + \lambda \left(\sum_{\mathbf{z}} q(\mathbf{z}|\mathbf{x}) - 1 \right) \right] &= 0 \\ 946 \Rightarrow q^*(\mathbf{z}|\mathbf{x}) &= P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x}) \text{ and } \lambda = 1 \\ 947 \Rightarrow P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x}) &= \arg \min_{q(\mathbf{z}|\mathbf{x})} \text{Var}_{q(\mathbf{z}|\mathbf{x})} \left[\frac{\pi_\theta(\mathbf{z}|\mathbf{x})}{q(\mathbf{z}|\mathbf{x})} \pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}) \right]. \end{aligned} \quad (33)$$

948
949
950

951 Therefore, we show that optimizing the variational posterior q_ϕ to approximate the true posterior
952 $P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x})$ in Eq. (9) naturally yields an (approximately) optimal behavior policy for RL, one that
953 minimizes the variance of the objective estimator. In practice, the trained variational posterior q_ϕ
954 can thus be employed as the behavior policy to reduce variance, which is fully compatible with
955 actor-critic frameworks that incorporate advantage estimation with baselines.
956957 A.9 CONNECTION TO REINFORCEMENT LEARNING TEACHERS
958959 In the derivation of our method, the ELBO objective in Eq. (4) minimizes *reverse* KL divergence
960 $\mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}') || P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x}))$. As analyzed in Section 2.3, we propose to optimize $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')$
961 using the *forward* KL divergence $\mathbb{D}_{\text{KL}}(P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x}) || q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}'))$, which shares the same optimal
962 solution.963 Alternatively, we can also optimize the reverse KL divergence by policy gradient method as follows:
964

965
$$\begin{aligned} \nabla_\phi \mathcal{L}_{\text{ELBO}} &= \nabla_\phi \mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}') || P_\theta(\mathbf{z}|\mathbf{x}, \mathcal{Y}_\mathbf{x})) \\ 966 &= \nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')} [\log \pi_\theta(\mathbf{z}, \mathcal{Y}_\mathbf{x}|\mathbf{x}) - \log q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')] \\ 967 &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')} \left[\log \pi_\theta(\mathbf{z}, \mathcal{Y}_\mathbf{x}|\mathbf{x}) \nabla_\phi \log q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}') - \log q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}') \nabla_\phi \log q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}') \right] \\ 968 &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')} \left[\underbrace{\left(\log \pi_\theta(\mathcal{Y}_\mathbf{x}|\mathbf{x}, \mathbf{z}) - \log \frac{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}')}{\pi_\theta(\mathbf{z}|\mathbf{x})} \right)}_{\text{reward}} \nabla_\phi \log q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}') \right]. \end{aligned} \quad (34)$$

969
970
971

972 More concisely, the reverse KL divergence can be alternatively minimized via reinforcement learning
 973 using $\log \pi_\theta(\mathcal{Y}_x | \mathbf{x}, \mathbf{z}) - \log \frac{q_\phi(\mathbf{z} | \mathbf{x}, \mathbf{y}')}{\pi_\theta(\mathbf{z} | \mathbf{x})}$ as the reward function.
 974

975 This derivation establishes a connection to Reinforcement Learning Teachers (RLTs) (Cetin et al.,
 976 2025), who focus on training reasoning LLMs to act as teachers for distilling new students. Their
 977 approach introduces RLTs optimized specifically for effective student distillation. RLTs are trained
 978 by GRPO using dense rewards obtained by feeding each explanation to the student and evaluating its
 979 understanding of the solution.

980 Specifically, the dense reward in RLT combines two components: one measuring the student's
 981 likelihood of reaching the correct solution (analogous to $\log \pi_\theta(\mathcal{Y}_x | \mathbf{x}, \mathbf{z})$), and another regularizing
 982 the teacher's explanation to remain coherent from the student's perspective given only its prior
 983 knowledge and the question (analogous to $-\log \frac{q_\phi(\mathbf{z} | \mathbf{x}, \mathbf{y}')}{\pi_\theta(\mathbf{z} | \mathbf{x})}$).
 984

985 While RLT employs an intuitively designed reward, our work provides rigorous theoretical justification
 986 from a variational inference perspective. Furthermore, we enhance the method with a tighter IWAE-
 987 style lower bound and an accuracy-based estimator, as detailed in Section 2.2.

988 **B RELATED WORK**
 989

990 **SFT and RL methods for reasoning.** Reasoning has emerged as a central capability of LLMs,
 991 driving advances in domains such as mathematics, programming, and scientific discovery (Jaech et al.,
 992 2024; Comanici et al., 2025; Team et al., 2025). Among the approaches developed to strengthen these
 993 abilities, SFT and RL have become the two dominant paradigms (Uesato et al., 2022; Rafailov et al.,
 994 2023; Guha et al., 2025; Hu et al., 2025; Hochlehnert et al., 2025). Building on the DeepSeek-R1
 995 framework (Shao et al., 2024; Guo et al., 2025), a range of new RL algorithms have been proposed,
 996 including Dr. GRPO (Liu et al., 2025), DAPO (Yu et al., 2025a), REINFORCE++ (Hu, 2025),
 997 VinePPO (Kazemnejad et al., 2024), and VAPO (Yue et al., 2025b). In parallel, extensive empirical
 998 studies have explored the design space of RL for reasoning (Zeng et al., 2025; Team et al., 2025),
 999 focusing on dimensions such as curriculum learning (Wen et al., 2025; Luo et al., 2025) and reward
 1000 design (Gao et al., 2024; Cui et al., 2025a; Ma et al., 2023; Qi et al., 2025). While early progress
 1001 has centered on mathematical reasoning, recent work has extended RL-based methods to code and
 1002 software engineering tasks (Liu & Zhang, 2025; Xie et al., 2025; Wei et al., 2025; Yang et al., 2025;
 1003 Chen et al., 2025; Li et al., 2025b), as well as to agentic problem-solving scenarios (Wang et al.,
 1004 2025; Jin et al., 2025; Jiang et al., 2025; Xue et al., 2025).

1005 **Decomposing thinking and answering processes.** Traditional studies on LLM reasoning ability
 1006 often treat model responses holistically. In contrast, a recent line of research explicitly decomposes
 1007 the LLM response into a thinking trace \mathbf{z} and a final answer \mathbf{y} , given a question \mathbf{x} (Chen et al., 2024;
 1008 Xiang et al., 2025; Zhou et al., 2025; Zhong et al., 2025). This decomposition offers several novel
 1009 and useful perspectives.

1010 Zhou et al. (2025) propose VeriFree, which directly optimizes $P_\theta(\mathbf{y} | \mathbf{x}) = \mathbb{E}_{\pi_\theta(\mathbf{z} | \mathbf{x})}[\pi_\theta(\mathbf{y} | \mathbf{x}, \mathbf{z})]$ using
 1011 policy gradient. Their algorithm simultaneously optimizes $\pi_\theta(\mathbf{z} | \mathbf{x})$ via policy gradient methods with
 1012 $\pi_\theta(\mathbf{y} | \mathbf{x}, \mathbf{z})$ as a reward, and performs weighted SFT on $\pi_\theta(\mathbf{y} | \mathbf{x}, \mathbf{z})$. This approach demonstrates
 1013 strong performance in general domains where rule-based verifiers are typically unavailable. Subse-
 1014 quent works (Yu et al., 2025b; Xu et al., 2025) further improve upon this by reshaping the reward, e.g.,
 1015 intuitively replacing the product of token probabilities with the mean when computing $\pi_\theta(\mathbf{y} | \mathbf{x}, \mathbf{z})$.

1016 Chen et al. (2024) introduce LaTRO, formulating reasoning as sampling from a latent distribution
 1017 $q(\mathbf{z} | \mathbf{x})$ and optimizing $\log P_\theta(\mathbf{y} | \mathbf{x})$ via a variational manner. Their derived lower bound is:

$$1018 \log P_\theta(\mathbf{y} | \mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z} | \mathbf{x})}[\log \pi_\theta(\mathbf{y} | \mathbf{x}, \mathbf{z})] - \mathbb{D}_{\text{KL}}(q(\mathbf{z} | \mathbf{x}) \parallel \pi_\theta(\mathbf{z} | \mathbf{x})).$$

1019 They set the proposal distribution $q(\mathbf{z} | \mathbf{x})$ to $\pi_\theta(\mathbf{z} | \mathbf{x})$, resulting in a reinforcement learning algorithm
 1020 where $\log \pi_\theta(\mathbf{y} | \mathbf{x}, \mathbf{z})$ serves as the reward. Tang et al. (2025) and Ruan et al. (2025) tighten this
 1021 bound using ideas similar to IWAE (Burda et al., 2015).

1022 A more natural choice for the variational distribution is the true posterior $P_\theta(\mathbf{z} | \mathbf{x}, \mathbf{y})$, though it is
 1023 intractable. Hoffman et al. (2023) use MCMC to sample from the posterior, while Hu et al. (2024)
 1024 employ GFlowNets (Bengio et al., 2023) to fine-tune an LLM to approximate it. Both methods use
 1025 an EM-like algorithm to optimize the ELBO of $\log P_\theta(\mathbf{y} | \mathbf{x})$.

1026 Our approach uses forward KL divergence to train a variational posterior and derives a novel objective
 1027 based on a tighter IWAE-style bound. Additionally, we propose an accuracy-based estimator for
 1028 $\pi_\theta(\mathcal{Y}_x | \mathbf{x}, \mathbf{z}_k)$, instead of the likelihood-based estimator used in Zhou et al. (2025). We also build
 1029 connections to other mainstream finetuning algorithms that enhance reasoning, such as RFT and GRPO.
 1030

1031 **Reinforcement learning as probabilistic inference.** Previous works have also explored connections
 1032 between reinforcement learning and probabilistic inference. Notably, Levine (2018) discuss RL
 1033 and control from a probabilistic inference perspective. Their approach begins from a reinforcement
 1034 learning standpoint, where the goal is to search for an optimal policy, and constructs a probabilistic
 1035 graphical model (PGM) in which the posterior distribution over actions corresponds to an optimal
 1036 policy. This is achieved by defining observations in the PGM based on rewards, and the resulting
 1037 inference problem is then solved via variational inference. This formulation differs from our approach.
 1038

1039 In contrast, our method starts from the objective of maximizing $\log P_\theta(\mathcal{Y}_x | \mathbf{x})$, treats the reasoning
 1040 process as a discrete latent variable \mathbf{z} , and optimizes the ELBO to train the reasoning model. We
 1041 subsequently draw connections between our framework and other RL algorithms (Section 3). From
 1042 an application perspective, our work focuses on the language domain and explicitly models reasoning
 1043 traces as discrete latent variables, which further distinguishes it from the aforementioned work.
 1044

C TRAINING DETAILS

1045 In this section, we detail the training procedure used in our method. Our framework builds on
 1046 LLaMA-Factory (Zheng et al., 2024). By default, SFT averages token-level cross-entropy over
 1047 all valid tokens in a batch. However, as shown in Section 2, our variables \mathbf{z} and \mathbf{y} are defined
 1048 at the sentence level. To align with this, we modify the objective: instead of normalizing by the
 1049 number of valid tokens, we sum the loss over all tokens and divide by a constant equal to the average
 1050 response length in the training set (precomputed offline). This change parallels the difference between
 1051 GRPO (Shao et al., 2024) and Dr. GRPO (Liu et al., 2025), thus we name this slight modification as
 1052 **Dr. SFT**. Both our models and Bespoke-Stratos-4B/8B † are trained with this modified objective. We
 1053 further extend the framework with weighted SFT, as the original LLaMA-Factory does not support
 1054 weighting. This feature is essential for parts of our method that require weighted training. All
 1055 experiments are conducted on NVIDIA H100 GPUs.
 1056

C.1 TRAINING THE INITIAL REASONING MODEL AND VARIATIONAL POSTERIOR

1057 For training, we use the following settings:
 1058

1. Initial reasoning model π_{θ_0} (following Labs (2025)):

- 1062 • batch_size=96, cutoff_len=16384
- 1063 • Optimizer: AdamW (Loshchilov & Hutter, 2019) with adam_beta1=0.9,
 1064 adam_beta2=0.999, adam_epsilon=1.0e-8, weight_decay=0
- 1065 • Learning rate schedule: cosine with warmup_ratio=0.1
- 1066 • learning_rate=1.0e-5, max_grad_norm=1.0
- 1067 • Training for 3 epochs
- 1068 • Precision: bfloat16
- 1069 • Baselines (Bespoke-Stratos-7B/32B and Bespoke-Stratos-4B/8B †) are trained with the
 1070 same setup

2. Variational posterior q_ϕ :

- 1074 • batch_size=16
- 1075 • Optimizer: AdamW with adam_beta1=0.9, adam_beta2=0.95,
 1076 adam_epsilon=1.0e-8, weight_decay=1.0e-4
- 1077 • warmup_ratio=0.05
- 1078 • Training for 10 epochs
- 1079 • All other hyperparameters follow those used for π_{θ_0}

1080
1081

C.2 DETAILS OF SAMPLING FROM VARIATIONAL POSTERIOR

1082
1083
1084
1085
1086
1087
1088

Using the trained variational posterior q_ϕ , we sample 8 reasoning traces (including final answers) for each question in Bespoke-Stratos-17k with vLLM (Kwon et al., 2023), using `temperature=0.7`, `top_p=1.0`, `top_k=-1`, `max_tokens=32764`, and `dtype=bfloat16`. After obtaining the sampled reasoning traces, we compute the importance weight $\tilde{\rho}_k$ for each question-thinking-answer triplet using the pre-trained initial reasoning model π_{θ_0} and the variational posterior q_ϕ . This is done in forward mode by evaluating the log-likelihoods under both models, without requiring backpropagation, which is efficient.

1089
1090
1091
1092
1093

To estimate the term $\pi_\theta(\mathcal{Y}_x | \mathbf{x}, \mathbf{z})$ used in $\tilde{\rho}_k$, we adopt an *accuracy-based estimator*. Specifically, for each question and each sampled thinking trace, we use π_θ to generate 8 answers under the same sampling configuration as above. The correctness of these answers is evaluated using the math/code verifiers from SkyThought, and the average accuracy is taken as the estimate of $\pi_\theta(\mathcal{Y}_x | \mathbf{x}, \mathbf{z})$. Experiments utilizing the accuracy-based estimator are labeled as “**-Acc**”.

1094
1095
1096
1097
1098
1099

Additionally, we employ the geometric mean of token-level probabilities under $\pi_\theta(\mathcal{Y}_x | \mathbf{x}, \mathbf{z})$ as an alternative and intuitive estimator. This approach mitigates the inherent length bias present in the strict definition of $\pi_\theta(\mathcal{Y}_x | \mathbf{x}, \mathbf{z})$, which computes the product of token probabilities and consequently assigns excessively small values to longer reasoning traces. This estimator provides an intuitive approximation without requiring an external verifier. Experiments utilizing the estimator based on geometric mean are labeled as “**-GML**”, while those with the naive estimator are labeled as “**-L**”.

1100
1101
1102

This process results in a weighted dataset where each sample consists of a question-thinking-answer triplet along with its corresponding weight $\tilde{\rho}_k$, which will be utilized in subsequent training stages.

1103
1104

C.3 DETAILS OF TRAINING FINAL REASONING MODEL

1105
1106
1107
1108
1109
1110

To train the final reasoning model π_θ , we adopt the following procedure. For the 17k data setting, we select, for each question, the reasoning trace with the highest importance weight $\tilde{\rho}_k$ among the 8 samples sampled from the variational posterior. In experiments using the accuracy-based estimator, we pair the selected reasoning trace with a randomly chosen verified answer generated by the initial reasoning model. For other estimators, the original answer from the dataset is retained. The resulting synthetic data is then mixed with the original Bespoke-Stratos-17k dataset.

1111
1112
1113
1114

We maintain the same training configuration as used for the initial reasoning model, with one exception: the batch size is increased to `batch_size=192`. This adjustment ensures that the total number of optimization steps remains consistent with baseline models (e.g., Bespoke-Stratos-32B), as the mixed dataset is twice the size of the original.

1115
1116
1117
1118
1119
1120

For the 1k data setting, the baseline model (e.g., Bespoke-Stratos-7B-1K[†]) is trained with the following configuration: We adopt `batch_size=16` and `cutoff_len=32768`. We use the AdamW optimizer (Loshchilov & Hutter, 2019) with parameters `adam_beta1=0.9`, `adam_beta2=0.95`, `adam_epsilon=1.0e-8`, and `weight_decay=1.0e-4`. A cosine learning rate schedule is applied with `warmup_ratio=0.1`, alongside a learning rate of `1.0e-5` and gradient clipping at `max_grad_norm=1.0`. Training is conducted for 5 epochs.

1121
1122
1123
1124
1125
1126

In our method under the 1k setting, we do not combine with the original dataset. Instead, we use all 8 reasoning traces, weighted by $\tilde{\rho}_k$, which is faithful to Algorithm 1. To match the number of optimization steps in the baseline, we proportionally adjust the batch size while keeping all other hyperparameters unchanged.

D DETAILS OF EVALUATION

1127
1128
1129
1130
1131
1132
1133

We conduct all evaluations using SkyThought, specifically at commit 0d190f1.³ Team (2025b) suggest avoiding greedy decoding for models with long thinking traces. Thus, responses are sampled from the models using `temperature=0.7` and `top_p=1.0`. A generous token budget of `max_tokens=38912` is allocated to accommodate lengthy outputs.

³<https://github.com/NovaSky-AI/SkyThought>

1134 To maximize reproducibility, we perform inference using `dtype=float32`, accepting a po-
1135 tential decrease in speed for improved consistency. For model parallelism, we configure
1136 `tensor_parallel_size=4` for 4B/7B/8B models and `tensor_parallel_size=8` for the
1137 32B models. We choose vLLM (Kwon et al., 2023) as the inference backend. For models based
1138 on Qwen2.5, we use `vllm==0.7.0`, while for Qwen3-based models, we use `vllm==0.8.4`.
1139 Although we anticipate that these version differences have negligible impact on evaluation accuracy,
1140 we document them here to ensure full reproducibility.

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

E PROMPT TEMPLATES

In Section 2, we abstractly introduced how we define the prompt patterns used in the reasoning model π_θ and variational posterior q_ϕ . In this section, we provide details of the prompt templates used in practice, as shown below.

Prompt template A (PA) for variational posterior q_ϕ

Your role as an assistant involves providing precise and accurate solutions before providing detailed explanations with your full work showing your systematic thinking process leading to each solution. Your explanations should show how you engaged in a comprehensive cycle of analysis, summarizing, exploration, reassessment, reflection, backtracing, and iteration to develop well-considered thinking process. Please structure your response into two main sections: Solution and Explanation. In the Solution section, present your well-thought solution that accurately answers the question. The solution should remain a logical, accurate, concise expression style and detail necessary step needed to reach the conclusion, formatted as follows:
 <|begin_of_solution|> {final formatted, precise, and clear solution} <|end_of_solution|>. In the Explanation section, comprehensively detail your reasoning process using the specified format: <|begin_of_explanation|> {explanation with steps separated with '\n\n'} <|end_of_explanation|> Each step should show detailed considerations leading to your solutions such as analysing questions, summarizing relevant findings, brainstorming new ideas, verifying the accuracy of the current steps, refining any errors, and revisiting previous steps.

Prompt template B (PB) for variational posterior q_ϕ

Your role as an assistant involves reconstructing the internal reasoning process that connects a provided question to its correct answer. Your task is to methodically reverse-engineer the logical steps, demonstrating a full cycle of analysis, summarization, idea generation, verification, error correction, and iterative refinement. Please structure your response into two distinct parts: Solution and Thought. In the Solution section, present the given correct answer in a precise and clear format:
 <|begin_of_solution|> {provided correct solution} <|end_of_solution|>. In the Thought section, articulate the step-by-step cognitive journey that leads to the solution. Use the specified format: <|begin_of_thought|> {detailed thought process with steps separated by '\n\n'} <|end_of_thought|>. Each step should reflect analytical breakdowns, synthesis of key points, generation of logical pathways, validation of each step's accuracy, refinement of any missteps, and reassessment of previous conclusions. The focus is solely on depicting the internal, structured thinking that arrives at the provided solution.

Prompt template for reasoning model π_θ

Your role as an assistant involves thoroughly exploring questions through a systematic long thinking process before providing the final precise and accurate solutions. This requires engaging in a comprehensive cycle of analysis, summarizing, exploration, reassessment, reflection, backtracing, and iteration to develop well-considered thinking process. Please structure your response into two main sections: Thought and Solution. In the Thought

1242
 1243 section, detail your reasoning process using the specified format:
 1244 <|begin_of_thought|> {thought with steps separated with '\\n\\n'}
 1245 <|end_of_thought|> Each step should include detailed
 1246 considerations such as analyzing questions, summarizing relevant
 1247 findings, brainstorming new ideas, verifying the accuracy of the
 1248 current steps, refining any errors, and revisiting previous steps.
 1249 In the Solution section, based on various attempts, explorations,
 1250 and reflections from the Thought section, systematically present
 1251 the final solution that you deem correct. The solution should
 1252 remain a logical, accurate, concise expression style and detail
 1253 necessary step needed to reach the conclusion, formatted as
 1254 follows: <|begin_of_solution|> {final formatted, precise, and
 1255 clear solution} <|end_of_solution|> Now, try to solve the
 1256
 1257
 1258
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1270
 1271
 1272
 1273
 1274
 1275
 1276
 1277
 1278
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288
 1289
 1290
 1291
 1292
 1293
 1294
 1295

1296 Table 6: Performance of models trained from **Qwen2.5-7B-Instruct** trained on Bespoke-Stratos-17k.
1297 The best and second-best results are highlighted using **bold text** and underlined text, respectively.
1298

1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 Method	1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 MATH500 Avg@2	Avg@2	AIME24 Avg@32	AIME25 Avg@32	AMC23 Avg@32	OlympiadBench Avg@2	Avg
Qwen2.5-7B-Instruct	75.60	10.94	7.40	51.10	39.91	36.99	
Bespoke-Stratos-7B	82.20	19.58	19.48	63.28	45.03	45.91	
RLT-7B	84.30	22.81	19.48	64.84	46.43	47.57	
Ours-PA-GML-7B	85.30	24.17	<u>20.42</u>	68.20	46.88	48.99	
Ours-PA-Acc-7B	83.40	22.50	20.83	65.39	<u>47.55</u>	47.94	
Ours-PB-GML-7B	84.00	22.08	<u>20.42</u>	<u>66.80</u>	46.29	47.92	
Ours-PB-Acc-7B	84.80	<u>23.96</u>	19.69	65.00	48.15	48.32	
Method	GPQA-D Avg@8	LCB-E Avg@8	LCB-M Avg@8	LCB-H Avg@8	MMLU-Pro Avg@1	Avg	
Qwen2.5-7B-Instruct	29.99	62.50	18.20	3.35	48.2	32.45	
Bespoke-Stratos-7B	39.02	69.30	23.06	2.95	60.59	38.98	
RLT-7B	41.60	72.32	25.06	3.66	61.28	40.78	
Ours-PA-GML-7B	43.62	74.52	27.79	4.78	60.88	42.32	
Ours-PA-Acc-7B	<u>43.56</u>	74.45	<u>29.43</u>	4.37	60.72	42.51	
Ours-PB-GML-7B	41.60	<u>74.73</u>	28.46	4.57	<u>61.14</u>	42.10	
Ours-PB-Acc-7B	41.67	75.07	29.49	<u>4.68</u>	61.28	<u>42.44</u>	

F EXTENDED RESULTS

F.1 EXTENDED MAIN RESULTS

Due to space constraints in the main paper, we present extended evaluation results in this section.

We report the evaluation results for models fine-tuned from Qwen2.5-7B-Instruct on the Bespoke-Stratos-7B dataset in Table 6. All variants of our method outperform all baselines in terms of average accuracy, demonstrating the superiority of the variational reasoning approach. Notably, the two prompt templates used for the variational posterior q_ϕ yield similar results, indicating that our method is robust to the choice of template. We attribute this robustness to the fact that the posterior is obtained by fine-tuning the model q_ϕ , rather than through prompt engineering alone, thereby reducing the sensitivity to specific prompt formulations.

Additionally, we plot the distributions of the thinking token length versus the log-likelihood ratio $\log \frac{\pi_\theta(\mathbf{z}_k | \mathbf{x})}{q_\phi(\mathbf{z}_k | \mathbf{x}, \mathbf{y}^*)}$ and the answer token length versus the log-likelihood of the answer $\log \pi_\theta(\mathcal{Y}_\mathbf{x} | \mathbf{x}, \mathbf{z}_k)$ in Figure 4. The results reveal strong correlations between these variables, indicating the presence of length biases. This observation further justifies the use of estimators based on accuracy or the geometric mean of token likelihood, rather than the naive likelihood.

Another noteworthy observation is that our evaluation results for General-Reasoner-4B (see Table 1) differ from those reported by Ma et al. (2025) in their original paper, despite using their officially released checkpoints and provided prompt template. This discrepancy can be attributed to several factors: (1) different evaluation frameworks: we employ SkyThought whereas they utilize simple-evals⁴; (2) different sampling configurations: we use `temperature=0.7` and `max_tokens=38912`, while they primarily employ greedy decoding (i.e., `temperature=0`) except for AIME24 and AIME25, along with a more constrained token budget of `max_tokens=8192`.

To facilitate a fair comparison, we provide results on the common benchmarks that we and Ma et al. (2025) both utilize, comparing our model accuracy with their officially reported values (see Table 7). As demonstrated, our method continues to outperform General-Reasoner-4B[‡] by a significant margin (where [‡] indicates their officially reported accuracy).

⁴<https://github.com/openai/simple-evals>

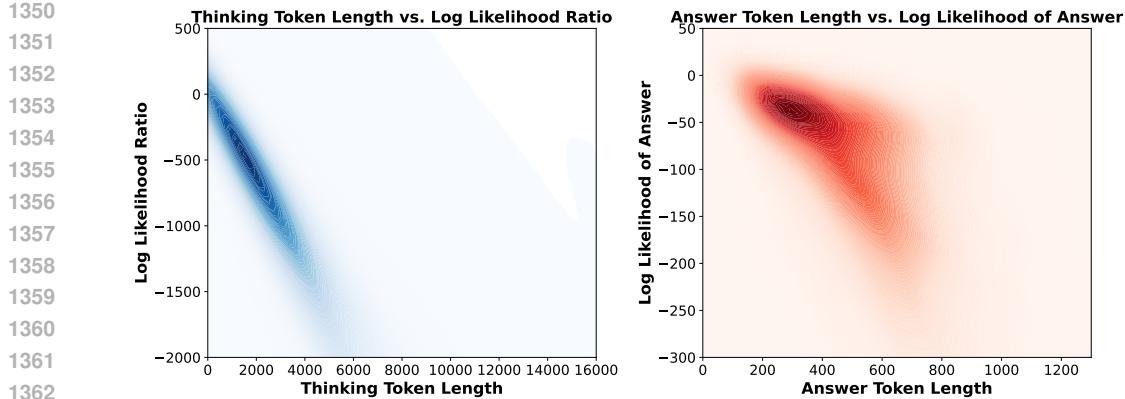


Figure 4: Density maps of the thinking token length versus the log-likelihood ratio $\log \frac{\pi_\theta(z_k | \mathbf{x})}{q_\phi(z_k | \mathbf{x}, \mathbf{y}')}$ (left), and the answer token length versus the log-likelihood of the answer $\log \pi_\theta(\mathcal{Y}_x | \mathbf{x}, z_k)$ (right).

Table 7: Performance of models trained from **Qwen3-4B-Base**. All models are trained on Bespoke-Stratos-17k except for General-Reasoner-4B. Here, † denotes accuracy values officially reported by [Ma et al. \(2025\)](#), rather than results obtained through our own evaluation. The best and second-best results are highlighted using **bold text** and underlined text, respectively.

Method	MATH500	AIME24	AIME25	AMC23
Qwen3-4B-Base	45.30	4.79	5.73	27.73
General-Reasoner-4B	71.70	19.06	16.77	55.00
General-Reasoner-4B †	80.6	20.0	15.4	60.0
Bespoke-Stratos-4B †	84.70	27.29	24.17	70.16
Ours-PB-GML-4B	<u>87.30</u>	33.54	<u>26.77</u>	<u>74.06</u>
Ours-PB-Acc-4B	88.30	<u>31.67</u>	27.29	75.63

Method	OlympiadBench	GPQA-D	MMLU-Pro
Qwen3-4B-Base	23.37	29.10	36.89
General-Reasoner-4B	45.18	40.97	61.36
General-Reasoner-4B †	47.7	42.9	62.8
Bespoke-Stratos-4B †	50.45	44.95	63.03
Ours-PB-GML-4B	<u>54.45</u>	45.52	<u>65.52</u>
Ours-PB-Acc-4B	55.71	<u>45.33</u>	65.53

F.2 EXTENDED ABLATION STUDIES

This section presents extended ablation studies analyzing: the impact of different training data sources and different ways of data usage (Appendix F.3); the effect of data overlap between variational posterior training and reasoning model training (Appendix F.4); the comparison between Dr. SFT and naive SFT (Appendix F.4); and the influence of cutoff length during training (Appendix F.6).

F.3 EFFECTS OF DIFFERENT FINAL DATA SOURCES AND WAYS OF DATA USAGE

In our main experiments, we prioritize training efficiency by using the 17k data setting, selecting the variational reasoning trace z_k with the highest weight $\tilde{\rho}_k$ and mixing it with the original Bespoke-Stratos-17k data (which results in double data sizes), rather than using all eight traces sampled from the variational posterior. To evaluate the impact of this simplification, we conduct ablation studies under the 1k data setting. We compare variants that either mix or do not mix with the original data, and that use either single-best trace selection (“-S”) or weighted multiple traces (“-M”).

Results are shown in Table 8. The best performance is achieved by the variant that uses weighted multiple reasoning traces without mixing with the original data. This suggests that, when computational cost is not a constraint, the optimal approach is to utilize all reasoning traces from the

Table 8: Ablation study on the effects of different final data sources (only sampled from variational posterior vs. mixed) and different ways to use samples from variational posterior to train reasoning models (single best reasoning trace selection (“-S”) vs. weighted multiple reasoning traces (“-M”)). This ablation is done in data 1k setting. The best and second-best results are highlighted using **bold text** and underlined text, respectively.

Method	MATH500	AIME24	AIME25	AMC23	Olympiad	Bench
	Avg@2	Avg@32	Avg@32	Avg@32	Avg@2	Avg
Qwen2.5-7B-Instruct	75.60	10.94	7.40	51.10	39.91	36.99
Bespoke-Stratos-7B-1K [†]	77.20	16.25	13.96	53.75	40.88	40.41
Ours-M-7B-1K	79.80	18.65	16.98	60.55	44.81	44.16
w/o Mix	81.30	<u>19.69</u>	18.44	61.64	45.99	45.41
Ours-S-7B-1K	<u>81.10</u>	19.90	17.08	60.39	43.92	<u>44.48</u>
w/o Mix	80.40	17.92	<u>17.81</u>	59.06	43.18	43.67
Method	GPQA-D	LCB-E	LCB-M	LCB-H	MMLU-Pro	Bench
	Avg@8	Avg@8	Avg@8	Avg@8	Avg@1	Avg
Qwen2.5-7B-Instruct	29.99	62.50	18.20	3.35	48.2	32.45
Bespoke-Stratos-7B-1K [†]	37.94	60.37	13.59	1.22	56.07	33.84
Ours-M-7B-1K	42.80	<u>65.73</u>	<u>18.99</u>	1.93	<u>60.49</u>	<u>37.99</u>
w/o Mix	<u>41.16</u>	68.13	21.42	1.42	60.94	38.62
Ours-S-7B-1K	40.34	64.84	18.20	<u>2.13</u>	59.54	37.01
w/o Mix	39.96	65.04	18.27	1.32	59.87	36.89

variational posterior, weighted by $\tilde{\rho}_k$, for training the final reasoning model. Another interesting observation is that for the single-trace method, data mixing improves performance, whereas for the weighted multi-trace method, mixing slightly degrades performance. This may indicate that the weighted ensemble of variational traces already provides sufficient information, making the original data redundant in this scenario.

F.4 EFFECTS OF DATA OVERLAP

In the 17k data setting, the variational posterior q_ϕ is trained on all 17k samples and generates thinking traces for the same set of 17k samples, which are subsequently used to train the final reasoning model π_θ . In other words, both the variational posterior and the final reasoning model are trained on the same set of question–answer pairs.

An interesting question is how our method performs in the absence of data overlap. To investigate this, we design two experimental settings: the first is the same 1k data setting introduced earlier; the second is constructed by splitting Bespoke-Stratos-17k into two non-overlapping subsets: one contains 15,710 samples (approximately 16k) and the other contains 1k samples. In the latter setting, we train the variational posterior q_ϕ and the initial reasoning model π_{θ_0} on the 16k subset. We then use the trained q_ϕ to sample thinking traces for the 1k subset, and employ both q_ϕ and π_{θ_0} to compute $\tilde{\rho}_k$. Finally, the final reasoning model π_θ is trained on the 1k subset using weighted multiple reasoning traces. This setting is referred to as “w/o Overlap”.

The results are presented in Table 9. Both the overlap and non-overlap variants exhibit similar performance in terms of average accuracy, and both outperform the baseline, Bespoke-Stratos-7B-1K[†]. This suggests that the trained variational posterior generalizes reasonably well and can be applied to broader scenarios.

F.5 COMPARING DR. SFT WITH NAIVE SFT

As detailed in Appendix C, we employ a slightly modified objective function, Dr. SFT, where the loss is defined as the sum of all valid token losses normalized by a constant, rather than the mean loss across valid tokens in the batch. We conduct an ablation study comparing this Dr. SFT approach against naive SFT when training the final reasoning model π_θ .

1458
 1459 Table 9: Ablation study on the effects of data overlap between variational posterior training and
 1460 reasoning model training. This ablation is done in data 1k setting. The best and second-best results
 1461 are highlighted using **bold text** and underlined text, respectively.

Method	MATH500	AIME24	AIME25	AMC23	OlympiadBench	Avg
	Avg@2	Avg@32	Avg@32	Avg@32	Avg@2	
Qwen2.5-7B-Instruct	75.60	10.94	7.40	51.10	39.91	36.99
Bespoke-Stratos-7B-1K [†]	77.20	16.25	13.96	<u>53.75</u>	40.88	40.41
Ours-7B-1K	79.80	<u>18.65</u>	<u>16.98</u>	60.55	44.81	44.16
w/o Mix	<u>81.30</u>	<u>19.69</u>	18.44	<u>61.64</u>	<u>45.99</u>	<u>45.41</u>
w/o Overlap	80.60	20.83	18.44	61.17	44.81	45.17
w/o Mix w/o Overlap	82.00	18.75	16.88	63.52	47.11	45.65

Method	GPQA-D	LCB-E	LCB-M	LCB-H	MMLU-Pro	Avg
	Avg@8	Avg@8	Avg@8	Avg@8	Avg@1	
Qwen2.5-7B-Instruct	29.99	62.50	18.20	3.35	48.2	32.45
Bespoke-Stratos-7B-1K [†]	37.94	60.37	13.59	1.22	56.07	33.84
Ours-7B-1K	42.80	65.73	18.99	1.93	60.49	37.99
w/o Mix	<u>41.16</u>	<u>68.13</u>	21.42	1.42	60.94	38.62
w/o Overlap	38.19	66.00	19.84	1.12	<u>60.65</u>	37.16
w/o Mix w/o Overlap	39.65	68.82	<u>19.96</u>	2.44	60.36	<u>38.25</u>

1480 The results are presented in Table 10. Both variants demonstrate comparable performance, with less
 1481 than 2% difference in average accuracy, and both outperform the baseline. This allows us to conclude
 1482 that the primary performance improvement stems from the variational reasoning mechanism rather
 1483 than from this minor modification to the objective function.

F.6 ABLATION STUDY ON THE EFFECTS OF CUTOFF LENGTH WHEN TRAINING

1484 In our main experiments (17k data setting), we use a cutoff length of `cutoff_len=16384`.
 1485 To investigate the impact of this hyperparameter, we conduct an ablation study comparing two
 1486 variants: one using the default `cutoff_len=16384` (denoted as “-Len16k”) and another with
 1487 `cutoff_len=32768` (denoted as “-Len32k”).

1488 The results are presented in Table 11. Both variants exhibit similar performance. We further analyze
 1489 the average completion token lengths on several evaluation benchmarks (see Table 12). All methods
 1490 produce significantly longer reasoning traces compared to the Qwen2.5-7B-Instruct, with our methods
 1491 generating slightly longer thinking traces. Notably, the 16k and 32k cutoff variants result in similar
 1492 generation lengths during inference. This indicates that increasing the cutoff length beyond 16k has a
 1493 minimal effect on the model’s output. Therefore, we can confidently use the 16k setting for better
 1494 training efficiency without sacrificing performance.

1495
 1496
 1497
 1498
 1499
 1500
 1501
 1502
 1503
 1504
 1505
 1506
 1507
 1508
 1509
 1510
 1511

1512
1513 Table 10: Ablation study comparing Dr. SFT and Naive SFT. This ablation is done in data 17k setting.
1514 The best and second-best results are highlighted using **bold text** and underlined text, respectively.

Method	MATH500 AIME24 AIME25 AMC23				OlympiadBench	Avg
	Avg@2	Avg@32	Avg@32	Avg@32	Avg@2	
Qwen2.5-7B-Instruct	75.60	10.94	7.40	51.10	39.91	36.99
Bespoke-Stratos-7B	82.20	19.58	19.48	63.28	45.03	45.91
RLT-7B	<u>84.30</u>	22.81	19.48	64.84	46.43	47.57
Ours-PA-GML-7B	85.30	24.17	20.42	68.20	46.88	48.99
w/ naive SFT	84.00	22.60	20.10	65.31	49.11	48.23
Ours-PA-Acc-7B	83.40	22.50	<u>20.83</u>	65.39	<u>47.55</u>	47.94
w/ naive SFT	84.10	<u>23.02</u>	21.04	67.66	46.96	<u>48.56</u>

Method	GPQA-D	LCB-E	LCB-M	LCB-H	MMLU-Pro	Avg
	Avg@8	Avg@8	Avg@8	Avg@8	Avg@1	
Qwen2.5-7B-Instruct	29.99	62.50	18.20	3.35	48.2	32.45
Bespoke-Stratos-7B	39.02	69.30	23.06	2.95	60.59	38.98
RLT-7B	41.60	72.32	25.06	3.66	<u>61.28</u>	40.78
Ours-PA-GML-7B	43.62	74.52	<u>27.79</u>	<u>4.78</u>	60.88	42.32
w/ naive SFT	42.49	<u>74.73</u>	25.85	3.66	61.41	41.63
Ours-PA-Acc-7B	<u>43.56</u>	74.45	29.43	4.37	60.72	42.51
w/ naive SFT	42.11	75.69	<u>27.79</u>	5.39	61.22	<u>42.44</u>

1534
1535 Table 11: Ablation study on effects of cutoff length used in training. This ablation is done in
1536 data 17k setting. Len16k: cutoff_len=16384; Len32k: cutoff_len=32768. The best and
1537 second-best results are highlighted using **bold text** and underlined text, respectively.

Method	MATH500 AIME24 AIME25 AMC23				OlympiadBench	Avg
	Avg@2	Avg@32	Avg@32	Avg@32	Avg@2	
Qwen2.5-7B-Instruct	75.60	10.94	7.40	51.10	39.91	36.99
Bespoke-Stratos-7B	82.20	19.58	19.48	63.28	45.03	45.91
RLT-7B	<u>84.30</u>	<u>22.81</u>	19.48	64.84	46.43	47.57
Ours-PA-GML-7B-Len16k	85.30	24.17	<u>20.42</u>	68.20	<u>46.88</u>	48.99
Ours-PA-GML-7B-Len32k	84.10	22.08	20.94	<u>66.80</u>	48.37	48.46

Method	GPQA-D	LCB-E	LCB-M	LCB-H	MMLU-Pro	Avg
	Avg@8	Avg@8	Avg@8	Avg@8	Avg@1	
Qwen2.5-7B-Instruct	29.99	62.50	18.20	3.35	48.2	32.45
Bespoke-Stratos-7B	39.02	69.30	23.06	2.95	60.59	38.98
RLT-7B	41.60	72.32	25.06	3.66	<u>61.28</u>	40.78
Ours-PA-GML-7B-Len16k	43.62	<u>74.52</u>	<u>27.79</u>	4.78	60.88	42.32
Ours-PA-GML-7B-Len32k	42.49	74.93	28.58	<u>4.37</u>	61.64	42.40

1553
1554 Table 12: Average completion token length of models trained from Qwen2.5-7B-Instruct.
1555

Method	MATH500	AIME24	AIME25	AMC23	MMLU-Pro
	Avg@2	Avg@32	Avg@32	Avg@32	Avg@1
Qwen2.5-7B-Instruct	564	1270	1027	849	531
Bespoke-Stratos-7B	5801	18413	<u>15769</u>	10921	3889
RLT-7B	5508	18143	<u>15769</u>	10986	3942
Ours-PA-GML-7B	5677	18299	16471	11338	3924
Ours-PA-GML-7B-Len32k	5688	18170	16747	11531	3965
Ours-PA-Acc-7B	5688	18170	16747	11531	3965
Ours-PB-GML-7B	5803	18479	16615	11080	4052
Ours-PB-Acc-7B	5787	18651	16696	11591	3974

1566
1567

F.7 EFFECTS OF WEIGHTS FOR TRAINING THE REASONING MODEL

1568
1569
1570
1571
1572
1573
1574
1575
1576

Our framework employs weighted supervised fine-tuning (SFT) for the reasoning model, as formalized in Eq. 6. The weight $\rho_k = \frac{\pi_\theta(z_k|x)}{q_\phi(z_k|x, y')} \cdot \pi_\theta(\mathcal{Y}_x|x, z_k)$ is derived from an IWAE-style evidence lower bound (ELBO). To analyze the contribution of the weighting scheme, we conduct an ablation study with two variants: (1) uniform weighting ($\rho_k = 1$), which reduces the method to standard SFT on traces sampled from the variational posterior; and (2) correctness-only weighting, which uses only the final answer probability $\pi_\theta(\mathcal{Y}_x|x, z_k)$ and omits the likelihood ratio. Results in Table 13 show that both components of the full weighting scheme contribute positively to performance. These findings align with our theoretical derivation and validate the design of the objective.

1577
1578
1579

Table 13: Ablation study on the effects of different SFT weights for training the reasoning model. The best and second-best results are highlighted using **bold text** and underlined text, respectively.

Method	MATH500	AIME24	AIME25	AMC23	OlympiadBench	Avg
	Avg@2	Avg@32	Avg@32	Avg@32	Avg@2	
Qwen3-4B-Base	45.30	4.79	5.73	27.73	23.37	21.38
General-Reasoner-4B	71.70	19.06	16.77	55.00	45.18	41.54
Bespoke-Stratos-4B [†]	84.70	27.29	24.17	70.16	50.45	51.35
Ours-PB-Acc-4B	88.30	31.67	27.29	75.63	55.71	55.72
w/ $\rho_k = 1$	87.00	31.04	26.04	72.89	52.52	53.90
w/ $\rho_k = \pi_\theta(\mathcal{Y}_x x, z_k)$	<u>88.20</u>	<u>31.45</u>	<u>26.56</u>	<u>73.43</u>	<u>54.45</u>	<u>54.82</u>

Method	GPQA-D	LCB-E	LCB-M	LCB-H	MMLU-Pro	Avg
	Avg@8	Avg@8	Avg@8	Avg@8	Avg@1	
Qwen3-4B-Base	29.10	18.54	5.46	1.32	36.89	18.26
General-Reasoner-4B	40.97	61.40	17.90	2.85	61.36	36.90
Bespoke-Stratos-4B [†]	44.95	71.22	19.54	3.25	63.03	40.40
Ours-PB-Acc-4B	45.33	80.29	33.68	5.79	65.53	46.12
w/ $\rho_k = 1$	<u>45.07</u>	78.09	28.82	5.38	64.24	44.32
w/ $\rho_k = \pi_\theta(\mathcal{Y}_x x, z_k)$	<u>45.07</u>	<u>78.43</u>	<u>30.27</u>	<u>5.48</u>	<u>64.41</u>	<u>44.73</u>

1596
1597

F.8 EFFECTS OF NOISY HINTS

1598
1599
1600
1601
1602
1603
1604
1605
1606
1607

As a sanity check, we conduct an experiment to confirm a basic expectation: sampling reasoning traces from the well-trained variational posterior should depend critically on the quality of the conditioning hint y' . We generate noisy hints by using an LLM (Qwen3-4B-Instruct-2507) to rewrite the original hints, introducing errors, and use these noisy hints for sampling reasoning traces using the variational posterior. The results, shown in Table 14, confirm that providing noisy hints significantly degrades the quality of the sampled reasoning traces. This outcome validates our basic assumption, as it demonstrates the sensitivity of the posterior to its conditioning input, which is consistent with its theoretical role.

1608
1609
1610

F.9 EFFECTS OF THE VERIFIER'S ACCURACY

1611
1612
1613
1614

To estimate $q_\phi(z_k|x, y')$ for the weight ρ_k , we have two options: a likelihood-based estimator (and its variant with a geometric mean modification) and an accuracy-based estimator. The likelihood-based approach does not require a verifier but relies on a reference answer, whereas the accuracy-based estimator depends on a verifier.

1615
1616
1617
1618
1619

To examine the robustness of our method to verifier accuracy, we conduct an ablation study using a simulated, highly inaccurate “dummy verifier” by setting all $q_\phi(z_k|x, y')$ values to 0.5. As shown in Table 15, while an inaccurate verifier can degrade performance, our method still outperforms the baselines. We attribute this robustness to the fact that the variational posterior is conditioned on a reasoning hint, which maintains a high overall correctness. Nevertheless, these results confirm that a more accurate verifier is still preferable in practice.

1620
 1621 Table 14: Ablation study on the effects of the quality of hints used for sampling from the variational
 1622 posterior. The best and second-best results are highlighted using **bold text** and underlined text,
 1623 respectively.

Method	MATH500	AIME24	AIME25	AMC23	OlympiadBench	Avg
	Avg@2	Avg@32	Avg@32	Avg@32	Avg@2	
Qwen3-4B-Base	45.30	4.79	5.73	27.73	23.37	21.38
General-Reasoner-4B	71.70	19.06	16.77	55.00	45.18	41.54
Bespoke-Stratos-4B [†]	84.70	<u>27.29</u>	24.17	<u>70.16</u>	50.45	51.35
Ours-PB-Acc-4B	88.30	31.67	27.29	75.63	55.71	55.72
w/ noisy hints	86.10	<u>27.29</u>	<u>24.27</u>	70.00	<u>50.52</u>	<u>51.64</u>

Method	GPQA-D	LCB-E	LCB-M	LCB-H	MMLU-Pro	Avg
	Avg@8	Avg@8	Avg@8	Avg@8	Avg@1	
Qwen3-4B-Base	29.10	18.54	5.46	1.32	36.89	18.26
General-Reasoner-4B	40.97	61.40	17.90	2.85	61.36	36.90
Bespoke-Stratos-4B [†]	44.95	71.22	19.54	3.25	63.03	40.40
Ours-PB-Acc-4B	45.33	80.29	33.68	5.79	65.53	46.12
w/ noisy hints	44.82	<u>74.24</u>	<u>26.21</u>	<u>4.98</u>	<u>64.18</u>	<u>42.88</u>

F.10 EFFECTS OF GEOMETRIC MEAN MODIFICATION ON LIKELIHOOD RATIO

As derived in Section 2.2, the weight ρ_k used for training the reasoning model in Eq. (6) can be decomposed as $\rho_k = \frac{\pi_\theta(z_k|x)}{q_\phi(z_k|x, y')} \cdot \pi_\theta(\mathcal{Y}_x|x, z_k)$, where the first term, $\frac{\pi_\theta(z_k|x)}{q_\phi(z_k|x, y')}$, is the likelihood ratio of the thinking trace z_k .

The standard likelihood ratio is unbounded, which can lead to high variance, and exhibits a clear length bias as demonstrated in Figure 4 (left). To address this, we heuristically introduce a geometric mean modification to the likelihood ratio, as defined in Eq. 8. While this modification introduces estimation bias, we evaluate its practical utility by comparing our method against a variant that removes this modification. Results presented in Table 16 indicate that the geometric mean is indeed beneficial. The principled variant (strictly derived from variational inference) performs slightly worse than our modified version, yet still surpasses all baseline methods, confirming the overall robustness of the framework.

F.11 COMPARISON AGAINST BASELINES WITH MATCHED COMPUTATIONAL BUDGET

Our method involves training both a variational posterior and a reasoning model, and also includes additional sampling and forward passes to compute token probabilities, which incurs extra computational cost. To further ensure a fair comparison, we scaled the compute of the baseline, Bespoke-Stratos-4B, to exceed the total GPU hours used by our framework. This is achieved by increasing both its training epochs and batch size. Consequently, the baseline’s total number of training tokens also largely exceeds that of our method. The results, presented in Table 17, indicate that even under a similar computational budget, our approach achieves better performance. This demonstrates the practical value of variational reasoning.

1674

1675

1676

1677

1678

1679

Table 15: **Ablation study on the effects of the verifier’s accuracy.** The best and second-best results are highlighted using **bold text** and underlined text, respectively.

Method	MATH500	AIME24	AIME25	AMC23	OlympiadBench	Avg
	Avg@2	Avg@32	Avg@32	Avg@32	Avg@2	
Qwen3-4B-Base	45.30	4.79	5.73	27.73	23.37	21.38
General-Reasoner-4B	71.70	19.06	16.77	55.00	45.18	41.54
Bespoke-Stratos-4B [†]	84.70	27.29	24.17	70.16	50.45	51.35
Ours-PB-Acc-4B	88.30	<u>31.67</u>	27.29	75.63	55.71	55.72
w/ dummy verifier	<u>88.00</u>	<u>31.25</u>	<u>26.45</u>	<u>74.21</u>	<u>53.64</u>	<u>54.71</u>

Method	GPQA-D	LCB-E	LCB-M	LCB-H	MMLU-Pro	Avg
	Avg@8	Avg@8	Avg@8	Avg@8	Avg@1	
Qwen3-4B-Base	29.10	18.54	5.46	1.32	36.89	18.26
General-Reasoner-4B	40.97	61.40	17.90	2.85	61.36	36.90
Bespoke-Stratos-4B [†]	44.95	71.22	19.54	3.25	63.03	40.40
Ours-PB-Acc-4B	45.33	80.29	33.68	5.79	65.53	46.12
w/ dummy verifier	<u>45.26</u>	<u>79.67</u>	<u>31.12</u>	<u>4.98</u>	<u>64.90</u>	<u>45.18</u>

1692

Table 16: **Ablation study on the effects of geometric mean modification of the likelihood ratio.** The best and second-best results are highlighted using **bold text** and underlined text, respectively.

Method	MATH500	AIME24	AIME25	AMC23	OlympiadBench	Avg
	Avg@2	Avg@32	Avg@32	Avg@32	Avg@2	
Qwen3-4B-Base	45.30	4.79	5.73	27.73	23.37	21.38
General-Reasoner-4B	71.70	19.06	16.77	55.00	45.18	41.54
Bespoke-Stratos-4B [†]	84.70	27.29	24.17	70.16	50.45	51.35
Ours-PB-Acc-4B	88.30	<u>31.67</u>	<u>27.29</u>	75.63	55.71	55.72
w/o geometric mean	<u>88.00</u>	32.60	27.60	<u>74.14</u>	<u>52.30</u>	<u>54.93</u>

Method	GPQA-D	LCB-E	LCB-M	LCB-H	MMLU-Pro	Avg
	Avg@8	Avg@8	Avg@8	Avg@8	Avg@1	
Qwen3-4B-Base	29.10	18.54	5.46	1.32	36.89	18.26
General-Reasoner-4B	40.97	61.40	17.90	2.85	61.36	36.90
Bespoke-Stratos-4B [†]	44.95	71.22	19.54	3.25	63.03	40.40
Ours-PB-Acc-4B	45.33	80.29	33.68	<u>5.79</u>	65.53	46.12
w/o geometric mean	<u>45.20</u>	<u>78.57</u>	<u>31.18</u>	5.99	<u>65.22</u>	<u>45.23</u>

1710

1711

Table 17: **Performance comparison under a similar computational budget.** The best and second-best results are highlighted using **bold text** and underlined text, respectively.

Method	MATH500	AIME24	AIME25	AMC23	OlympiadBench	Avg
	Avg@2	Avg@32	Avg@32	Avg@32	Avg@2	
Qwen3-4B-Base	45.30	4.79	5.73	27.73	23.37	21.38
Bespoke-Stratos-4B [†]	84.70	27.29	24.17	70.16	50.45	51.35
w/ scaled compute	86.20	29.79	25.52	72.81	52.30	53.32
Ours-PB-GML-4B	<u>87.30</u>	33.54	<u>26.77</u>	<u>74.06</u>	<u>54.45</u>	<u>55.23</u>
Ours-PB-Acc-4B	88.30	<u>31.67</u>	27.29	75.63	55.71	55.72

Method	GPQA-D	LCB-E	LCB-M	LCB-H	MMLU-Pro	Avg
	Avg@8	Avg@8	Avg@8	Avg@8	Avg@1	
Qwen3-4B-Base	29.10	18.54	5.46	1.32	36.89	18.26
Bespoke-Stratos-4B [†]	44.95	71.22	19.54	3.25	63.03	40.40
w/ scaled compute	45.01	75.61	27.91	4.57	63.99	43.42
Ours-PB-GML-4B	45.52	<u>79.53</u>	<u>31.25</u>	6.20	<u>65.52</u>	<u>45.60</u>
Ours-PB-Acc-4B	<u>45.33</u>	80.29	33.68	<u>5.79</u>	65.53	46.12

1728 **G EXAMPLES OF GENERATED RESPONSES**
17291730
1731 In this section, we provide examples of responses from the variational posterior q_ϕ and the final
1732 reasoning model π_θ trained based on Qwen3-4B-Base. The provided example of the variational
1733 posterior q_ϕ is based on the variant with prompt template B, and the question-answer pair is one
1734 of the samples from the Bespoke-Stratos-17k dataset. The provided example of the final reasoning
1735 model π_θ is based on one of the samples in MATH500. See the provided examples below.
17361737 **Example of variational posterior q_ϕ** 1738 **Prompt:**

```

1739 <|im_start|>system
1740 Your role as an assistant involves reconstructing the internal
1741 reasoning process that connects a provided question to its correct
1742 answer. Your task is to methodically reverse-engineer the logical
1743 steps, demonstrating a full cycle of analysis, summarization, idea
1744 generation, verification, error correction, and iterative
1745 refinement. Please structure your response into two distinct
1746 parts: Solution and Thought. In the Solution section, present the
1747 given correct answer in a precise and clear format:
1748 <|begin_of_solution|> {provided correct solution}
1749 <|end_of_solution|>. In the Thought section, articulate the
1750 step-by-step cognitive journey that leads to the solution. Use the
1751 specified format: <|begin_of_thought|> {detailed thought process
1752 with steps separated by '\n\n'} <|end_of_thought|>. Each step
1753 should reflect analytical breakdowns, synthesis of key points,
1754 generation of logical pathways, validation of each step's
1755 accuracy, refinement of any missteps, and reassessment of previous
1756 conclusions. The focus is solely on depicting the internal,
1757 structured thinking that arrives at the provided
1758 solution.<|im_end|>
1759 <|im_start|>user
1760 Return your final response within \boxed{}. A figure is an
1761 equiangular parallelogram if and only if it is a
1762 $\text{rectangle} \quad \text{quad} \quad \text{regular polygon} \quad \text{quad} \quad \text{rhombus} \quad \text{quad} \quad \text{trapezoid} $<|im_end|>
1763 <|im_start|>assistant
1764 <|begin_of_solution|>
1765
1766 To determine the correct answer, we need to analyze the
1767 definitions and properties of the given options in relation to an
1768 equiangular parallelogram.
1769
1770 1. Definitions and Properties:
1771 - A parallelogram is a quadrilateral with both pairs of
1772 opposite sides parallel.
1773 - Equiangular means all angles are equal. In a quadrilateral,
1774 this means each angle is 90 degrees (since the sum of angles in a
1775 quadrilateral is 360 degrees, and  $360/4 = 90$ ).
1776 - A rectangle is a parallelogram with all angles equal to 90
1777 degrees.
1778 - A regular polygon (for quadrilaterals) is a square, which is
1779 both equiangular and equilateral.
1780 - A rhombus is a parallelogram with all sides equal, but not
1781 necessarily equiangular.
1782 - A square is a special case of both a rectangle and a
1783 rhombus, being both equiangular and equilateral.
1784 - A trapezoid has at least one pair of parallel sides, but not
1785 necessarily two pairs.
1786
1787 2. Analysis:

```

1782
 1783 - An equiangular parallelogram must have all angles equal to 90
 1784 degrees, making it a rectangle.
 1785 - The "if and only if" condition requires that the converse is
 1786 also true: a rectangle is an equiangular parallelogram.
 1787 - A regular polygon (quadrilateral) is a square, which is too
 1788 specific since the problem does not mention sides being equal.
 1789 - A rhombus does not require all angles to be equal, so it is not
 1790 necessarily equiangular.
 1791 - A square is a special case of a rectangle but requires both
 1792 equiangular and equilateral properties, which are not both
 1793 specified here.
 1794 - A trapezoid does not necessarily have two pairs of parallel
 1795 sides, hence not a parallelogram.

1796 3. **Conclusion**:

1797 - The correct answer is a rectangle, as it satisfies both the
 1798 equiangular condition and the properties of a parallelogram.

1799 \boxed{A}

1800 <|end_of_solution|>

1801 <|begin_of_thought|>

1802 **Response:**

1803 Okay, let me try to figure out this problem. The question is: "A
 1804 figure is an equiangular parallelogram if and only if it is a..."
 1805 with options given. The answer choices are rectangle, regular
 1806 polygon, rhombus, square, trapezoid. Hmm. So, first, let's break
 1807 down the terms here.

1808 An equiangular parallelogram. Let's recall definitions. A
 1809 parallelogram is a quadrilateral with both pairs of opposite sides
 1810 parallel. Equiangular means that all the angles are equal. In a
 1811 quadrilateral, if all angles are equal, each angle must be 90
 1812 degrees because the sum of angles in a quadrilateral is 360
 1813 degrees. So, 360 divided by 4 is 90. So, an equiangular
 1814 quadrilateral has four right angles. Wait, but a parallelogram by
 1815 definition already has opposite angles equal. So, if a
 1816 parallelogram is equiangular, that forces all angles to be 90
 1817 degrees, right? So, that would make it a rectangle. Because a
 1818 rectangle is a parallelogram with all angles equal to 90 degrees.
 1819 So, then, an equiangular parallelogram is a rectangle. Then the
 1820 "if and only if" part. So, the statement is saying that a figure
 1821 is an equiangular parallelogram exactly when it is a [answer]. So,
 1822 the converse would also have to be true. That is, if it's a
 1823 [answer], then it's an equiangular parallelogram, and vice versa.

1824 So, let's check the options. Option A is rectangle. If a figure is
 1825 a rectangle, then it's a parallelogram with all angles equal (to
 1826 90 degrees), so it's an equiangular parallelogram. Conversely, if
 1827 it's an equiangular parallelogram, then as we saw, all angles are
 1828 90 degrees, so it's a rectangle. So, that seems to satisfy the "if
 1829 and only if" condition.

1830 But wait, let's check the other options just to be thorough.
 1831 Option B is regular polygon. A regular polygon is a polygon that
 1832 is both equiangular and equilateral. In the case of a
 1833 quadrilateral, a regular polygon would be a square. So, a square
 1834 is a regular quadrilateral. But a parallelogram that is
 1835 equiangular is a rectangle, which isn't necessarily a square
 unless it's also equilateral (all sides equal). So, regular
 polygon is too specific here. Moreover, the question is about a

1836
 1837 parallelogram, which is a quadrilateral, but "regular polygon"
 1838 could refer to any regular polygon, not necessarily a
 1839 quadrilateral, but in the answer choices, it's listed as an
 1840 option. But since a regular polygon (quadrilateral) is a square,
 1841 but the question is about an equiangular parallelogram, which is a
 1842 rectangle. So, unless rectangles are considered regular polygons,
 1843 but rectangles are not regular polygons unless they are squares.
 1844 So, regular polygon is not the correct answer here. So, B is out.
 1845
 1846 Option C is rhombus. A rhombus is a parallelogram with all sides
 1847 equal. A rhombus doesn't have to be equiangular. In fact, a
 1848 rhombus can have angles of, say, 60 and 120 degrees. So, a rhombus
 1849 is not necessarily equiangular. So, an equiangular parallelogram
 1850 is a rectangle, not necessarily a rhombus. So, C is out.
 1851
 1852 Option D is square. A square is a parallelogram that is both
 1853 equiangular (all angles 90 degrees) and equilateral (all sides
 1854 equal). So, the problem here is that an equiangular parallelogram
 1855 is a rectangle, which is only a square if the sides are equal. But
 1856 the question doesn't specify that the sides are equal. So, an
 1857 equiangular parallelogram is a rectangle, but a rectangle is a
 1858 square only if it's equilateral. Since the problem is only
 1859 specifying equiangular (all angles equal) and parallelogram, the
 1860 figure is a rectangle, which is a type of parallelogram but not
 necessarily a square. Therefore, the "if and only if" would
 require that being a square is equivalent to being an equiangular
 parallelogram. But that's not true because a square is a special
 case. So, D is not correct.
 1861
 1862 Option E is trapezoid. A trapezoid is a quadrilateral with at
 1863 least one pair of parallel sides. However, a parallelogram has two
 1864 pairs of parallel sides, so a parallelogram is a special case of a
 1865 trapezoid (depending on the definition; in some definitions,
 1866 trapezoids have exactly one pair). But even so, a trapezoid isn't
 1867 necessarily a parallelogram. Moreover, an equiangular trapezoid
 1868 would have all angles equal, but in a trapezoid, which is not
 1869 necessarily a parallelogram, having all angles equal doesn't
 1870 guarantee that the sides are parallel. For example, an isosceles
 1871 trapezoid can be equiangular (with two pairs of equal angles), but
 1872 that's a different case. However, the question specifies an
 1873 equiangular parallelogram. So, if it's a parallelogram, then it's
 1874 already a quadrilateral with two pairs of parallel sides, and if
 1875 it's equiangular, all angles are 90 degrees, making it a
 rectangle. A trapezoid isn't required to have two pairs of
 parallel sides, so an equiangular trapezoid isn't necessarily a
 parallelogram. Therefore, E is out.
 1876
 1877 So, the only remaining option is A, rectangle. Let me just verify
 1878 again. If it's a rectangle, then it's a parallelogram with all
 1879 angles equal, so equiangular. Conversely, if it's an equiangular
 1880 parallelogram, all angles are 90 degrees, hence a rectangle. The
 "if and only if" holds here. So, the answer should be A.
 1881
 1882 Wait, but hold on. The problem says "a figure is an equiangular
 1883 parallelogram if and only if it is a [answer]". So, this is a
 1884 biconditional statement. So, the statement and the statement "it
 1885 is a [answer]" must be equivalent. Therefore, the [answer] must be
 1886 such that every equiangular parallelogram is a [answer], and every
 1887 [answer] is an equiangular parallelogram.
 1888
 1889 But as I thought earlier, a rectangle is an equiangular
 parallelogram. However, the reverse - is every rectangle an
 equiangular parallelogram? Well, yes, because a rectangle by

1890
 1891 definition has four right angles, so it's equiangular. So, the
 1892 equivalence holds. Therefore, the if and only if is satisfied.
 1893
 1894 But wait, the answer options include square. So, maybe there's a
 1895 confusion here. Let me think again. The question is asking for the
 1896 figure that is an equiangular parallelogram if and only if it is a
 1897 [answer]. So, the figure is a parallelogram that is equiangular,
 1898 and this happens exactly when it's a [answer]. So, since we
 1899 concluded that this [answer] is rectangle, then A is correct.
 1900
 1901 But let me check if any of the other options could also satisfy
 1902 the biconditional. For example, square. So, is a square an
 1903 equiangular parallelogram? Yes, because a square is a
 1904 parallelogram with all sides equal and all angles equal (90
 1905 degrees). However, the problem states "if and only if". So, the
 1906 figure is an equiangular parallelogram exactly when it is a
 1907 square. But that's not true. Because a rectangle is also an
 1908 equiangular parallelogram, but a square is a special case of a
 1909 rectangle. So, if the statement was "if and only if it is a
 1910 square", then it would be too restrictive, because there are
 1911 rectangles that are equiangular parallelograms which are not
 1912 squares. Therefore, the "if and only if" with square would not
 1913 hold. Similarly, regular polygon is too broad or specific. So, the
 1914 correct answer is rectangle.
 1915
 1916 Therefore, the answer should be A. But wait, the original problem
 1917 says "equiangular parallelogram". Wait, in some contexts, a
 1918 parallelogram being equiangular is the same as being a rectangle,
 1919 so equiangular parallelogram = rectangle. Then the question is
 1920 asking, "An equiangular parallelogram is a [answer] if and only if
 1921 it is a..." Hmm, perhaps the wording is confusing. Let me parse
 1922 the sentence again.
 1923
 1924 "A figure is an equiangular parallelogram if and only if it is a
 1925 [answer]"
 1926
 1927 So, the figure is defined as an equiangular parallelogram, and
 1928 this is equivalent (if and only if) it is a [answer]. So, the
 1929 definition of an equiangular parallelogram is "it is a [answer]".
 1930 Therefore, the [answer] must be the name given to the figure that
 1931 is exactly the equiangular parallelogram. Since we've concluded
 1932 that an equiangular parallelogram is a rectangle, then the
 1933 definition would be "a rectangle is an equiangular parallelogram".
 1934 But the answer options have rectangle as A. However, in logic, the
 1935 if and only if is symmetric. So, saying "a figure is an
 1936 equiangular parallelogram if and only if it is a rectangle" is
 1937 correct. So, that would be answer A.
 1938
 1939 But then why is square an option? Maybe I made a mistake here.
 1940 Let's consider that perhaps the question is from a country where
 1941 parallelogram definitions differ. For example, in some
 1942 definitions, a parallelogram is a quadrilateral with both pairs of
 1943 opposite sides parallel, and in those cases, a parallelogram is
 1944 only a rectangle if it has right angles. But in standard
 1945 definitions, a rectangle is a parallelogram with four right
 1946 angles, so that's correct.
 1947
 1948 Alternatively, maybe there is a different interpretation. Wait,
 1949 the problem is in the "if and only if" part. So, if we take
 1950 "equiangular parallelogram" as the definition, then the "if and
 1951 only if" it is a [answer]. So, the [answer] must be co-extensive
 1952 with the equiangular parallelogram. In other words, the set of
 1953 equiangular parallelograms is the same as the set of [answer]. So,

1944
 1945 since we know that an equiangular parallelogram is a rectangle,
 1946 but a rectangle is not necessarily a square (since rectangles can
 1947 have sides of different lengths), then the set of equiangular
 1948 parallelograms is exactly the set of rectangles. Therefore, the
 1949 answer is rectangle.
 1950
 1951 But maybe the problem is trying to trick us into thinking that a
 1952 square is the answer because it's a special case. But since the
 1953 problem doesn't mention sides being equal, just angles being
 1954 equal, it's a rectangle. So, the answer should be A.
 1955
 1956 Alternatively, perhaps there's a different approach. Let's recall
 1957 that in a parallelogram, opposite angles are equal. If it's
 1958 equiangular, then all angles are equal. So, in a parallelogram,
 1959 this forces all angles to be 90 degrees. Therefore, it's a
 1960 rectangle. Therefore, equiangular parallelogram is another name
 1961 for rectangle. So, if the question is asking for the term that
 1962 describes such a figure, the answer is rectangle. So, A.
 1963
 1964 But maybe the options are labeled differently. Let me check the
 1965 options again:
 1966
 1967 A) rectangle
 1968
 1969 B) regular polygon
 1970
 1971 C) rhombus
 1972
 1973 D) square
 1974
 1975 E) trapezoid
 1976
 1977 Yes. So, given that, the answer is A. But let me verify once more.
 1978 Suppose someone thinks that "equiangular" refers to both angles
 1979 and sides. But no, equiangular just means all angles are equal.
 1980 Equilateral would mean all sides are equal. So, the problem is
 1981 only mentioning angles. So, a parallelogram that is equiangular
 1982 must have all angles 90 degrees, hence a rectangle. Therefore, the
 1983 answer is A.
 1984
 1985 I think that's solid. So, I think the answer is A, rectangle.
 1986
 1987 ****Final Answer****
 1988 \boxed{A}

1983 Example of reasoning model π_0
 1984
Prompt:
 1985
 1986 <|im_start|>system
 1987 Your role as an assistant involves thoroughly exploring questions
 1988 through a systematic long thinking process before providing the
 1989 final precise and accurate solutions. This requires engaging in a
 1990 comprehensive cycle of analysis, summarizing, exploration,
 1991 reassessment, reflection, backtracing, and iteration to develop
 1992 well-considered thinking process. Please structure your response
 1993 into two main sections: Thought and Solution. In the Thought
 1994 section, detail your reasoning process using the specified format:
 1995 <|begin_of_thought|> {thought with steps separated with '\n\n'}
 1996 <|end_of_thought|> Each step should include detailed
 1997 considerations such as analysing questions, summarizing relevant
 1998 findings, brainstorming new ideas, verifying the accuracy of the
 1999 current steps, refining any errors, and revisiting previous steps.

1998
 1999 In the Solution section, based on various attempts, explorations,
 2000 and reflections from the Thought section, systematically present
 2001 the final solution that you deem correct. The solution should
 2002 remain a logical, accurate, concise expression style and detail
 2003 necessary step needed to reach the conclusion, formatted as
 2004 follows: <|begin_of_solution|> {final formatted, precise, and
 2005 clear solution} <|end_of_solution|> Now, try to solve the
 2006 following question through the above guidelines:<|im_end|>
 2007 <|im_start|>user
 2008 If $f(x) = \frac{3x-2}{x-2}$, what is the value of $f(-2) + f(-1) + f(0)$? Express your answer as a common fraction.<|im_end|>
 2009 <|im_start|>assistant
 2010 <|begin_of_thought|>
2011 Response:
 2012 Okay, so I need to find the value of $f(-2) + f(-1) + f(0)$ for the
 2013 function $f(x) = (3x - 2)/(x - 2)$. Hmm, let me start by recalling
 2014 how to evaluate a function at a specific point. For each x-value
 2015 given, I substitute it into the function and simplify. Let me
 2016 break this down step by step.
 2017
 2018 First, let's compute $f(-2)$. Substituting $x = -2$ into the function:
 2019 $f(-2) = [3*(-2) - 2]/[(-2) - 2] = [(-6) - 2]/[-4] = (-8)/(-4) = 2$.
 2020 Wait, that seems straightforward. The numerator becomes -8 and the
 2021 denominator becomes -4 , and a negative divided by a negative is
 2022 positive 2 . Okay, so $f(-2)$ is 2 .
 2023
 2024 Next, let's find $f(-1)$. Substituting $x = -1$:
 2025 $f(-1) = [3*(-1) - 2]/[(-1) - 2] = [(-3) - 2]/[-3] = (-5)/(-3) = \frac{5}{3}$.
 2026 Hmm, negative divided by negative is positive, so $\frac{5}{3}$. That
 2027 makes sense.
 2028
 2029 Now, moving on to $f(0)$. Substitute $x = 0$:
 2030 $f(0) = [3*0 - 2]/[0 - 2] = [0 - 2]/[-2] = (-2)/(-2) = 1$. Okay, so
 2031 $f(0)$ is 1 .
 2032
 2033 Now, adding these three results together: $2 + \frac{5}{3} + 1$. Let me
 2034 convert the whole numbers to fractions to make addition easier. 2
 2035 is $\frac{6}{3}$, and 1 is $\frac{3}{3}$. So adding them up:
 2036 $\frac{6}{3} + \frac{5}{3} + \frac{3}{3} = (6 + 5 + 3)/3 = \frac{14}{3}$.
 2037
 2038 Wait, let me double-check each calculation to make sure I didn't
 2039 make any mistakes. Starting with $f(-2)$:
 2040 Numerator: $3*(-2) = -6$, minus 2 is -8 . Denominator: $-2 - 2 = -4$.
 2041 So $-8/-4$ is indeed 2 . Correct.
 2042
 2043 For $f(-1)$:
 2044 Numerator: $3*(-1) = -3$, minus 2 is -5 . Denominator: $-1 - 2 = -3$.
 2045 So $-5/-3$ is $\frac{5}{3}$. Correct.
 2046
 2047 For $f(0)$:
 2048 Numerator: $3*0 = 0$, minus 2 is -2 . Denominator: $0 - 2 = -2$. So
 2049 $-2/-2$ is 1 . Correct.
 2050
 2051

```

2052
2053 Adding them up:  $2 + \frac{5}{3} + 1$ . Converting 2 and 1 to thirds: 2 is
2054  $\frac{6}{3}$ , 1 is  $\frac{3}{3}$ . So  $\frac{6}{3} + \frac{5}{3} + \frac{3}{3} = \frac{14}{3}$ . That seems right.
2055
2056 Alternatively, maybe I can add them in decimal form to verify. 2
2057 is 2.0,  $\frac{5}{3}$  is approximately 1.666..., and 1 is 1.0. Adding those:
2058  $2.0 + 1.666\ldots + 1.0 = 4.666\ldots$ , which is  $\frac{14}{3}$  (since 14 divided
2059 by 3 is approximately 4.666...). So that checks out.
2060
2061 I don't think I made any errors here. Each substitution was done
2062 correctly, and the arithmetic adds up. Therefore, the final answer
2063 should be  $\frac{14}{3}$ .
2064 <end_of_thought>
2065 <begin_of_solution>
2066 To find  $\left( f(-2) + f(-1) + f(0) \right)$  for  $f(x) = \frac{3x-2}{x-2}$ 
2067  $\backslash\backslash$ :
2068
2069 1. **Calculate  $\left( f(-2) \right)$ **  

2070  $\backslash[$   

2071  $f(-2) = \frac{3(-2) - 2}{-2 - 2} = \frac{-6 - 2}{-4} =$   

2072  $\frac{-8}{-4} = 2$   

2073  $\backslash]$ 
2074
2075 2. **Calculate  $\left( f(-1) \right)$ **  

2076  $\backslash[$   

2077  $f(-1) = \frac{3(-1) - 2}{-1 - 2} = \frac{-3 - 2}{-3} =$   

2078  $\frac{-5}{-3} = \frac{5}{3}$   

2079  $\backslash]$ 
2080
2081 3. **Calculate  $\left( f(0) \right)$ **  

2082  $\backslash[$   

2083  $f(0) = \frac{3(0) - 2}{0 - 2} = \frac{-2}{-2} = 1$   

2084  $\backslash]$ 
2085
2086 4. **Sum the results:**  

2087  $\backslash[$   

2088  $2 + \frac{5}{3} + 1 = \frac{6}{3} + \frac{5}{3} + \frac{3}{3} =$   

2089  $\frac{14}{3}$   

2090  $\backslash]$ 
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105

```