# CTRLS: Chain-of-Thought Reasoning via Latent State Transition

**Junda Wu**[1,*] **Yuxin Xiong**[1,*] **Xintong Li**[1] **Sheldon Yu**[1]
**Zhengmian Hu**[2] **Tong Yu**[2] **Rui Wang**[2] **Xiang Chen**[2] **Jingbo Shang**[1] **Julian McAuley**[1]

[1]University of California, San Diego [2]Adobe Research

## Abstract

Chain-of-thought (CoT) reasoning enables large language models (LLMs) to break down complex problems into explainable intermediate steps, significantly enhancing model transparency and performance in reasoning tasks. However, conventional CoT methods rely on heuristic sampling without structured modelling of reasoning transitions, constraining their ability to explore and discover diverse and effective reasoning trajectories. In this work, we introduce `CTRLS`, a framework that formulates CoT reasoning as a Markov decision process (MDP) with latent state transitions, enabling explainable and state-aware exploration via distributional reinforcement learning. By modelling reasoning actions as explicit probability distributions in latent space, our approach explicitly models epistemic uncertainty, facilitating robust exploration of the reasoning space. Enabled by our formulation, we propose an on-policy reinforcement learning scheme to iteratively refine latent transitions without fine-tuning of the underlying LLM. Theoretical analyses provide evidence lower bounds (ELBO), theoretically grounding our transition-aware modelling of latent reasoning dynamics.

## 1 INTRODUCTION

Chain-of-thought (CoT) reasoning has emerged as an effective paradigm for enabling large language models (LLMs) to tackle complex tasks by decomposing them into structured, interpretable intermediate reasoning steps (Wei et al., 2022; Kojima et al., 2022; Wu et al., 2024). However, conventional CoT prompting lacks transition-aware modelling, such that each step is generated autoregressively without capturing the underlying dynamics of reasoning transitions, limiting explainable exploration and diversity Yu et al. (2025); Zhang et al. (2025b); Gan et al. (2025); Hou et al. (2023), in reasoning trajectories (Wei et al., 2022; Kveton et al., 2025; Xia et al., 2025). On the other end, structured reasoning frameworks (e.g., Toolchain* (Zhuang et al.), program synthesis (Zhang et al., 2023), knowledge graph (Wu et al., 2025a)) enforce rigid intermediate steps via API calls or logic traces, offering structural control but sacrificing flexibility and semantic adaptability (Kojima et al., 2022; Wu et al., 2025a).

To illustrate the limitations of conventional CoT prompting, Figure 1 contrasts standard autoregressive reasoning with our transition-aware framework. On the left, traditional CoT unfolds step-by-step without modelling transitions between reasoning steps, often leading to premature commitment and limited trajectory diversity. In contrast, our method models reasoning as a stochastic trajectory over latent states, enabling explicit transition modelling and exploration of alternative reasoning paths. This comparison highlights the potential of transition-aware CoT to uncover more effective and diverse reasoning strategies.

To bridge the extremes, we propose transition-aware CoT reasoning `CTRLS`, a novel perspective that frames reasoning as structured trajectories within a latent state space. Each reasoning step corresponds to a continuous latent semantic state, with transitions dynamically learned via a latent-state MDP formulation. This modelling explicitly captures reasoning regularities and semantic relationships between intermediate steps, supporting explainable exploration.

Modelling transition-aware CoT reasoning presents fundamental challenges: (i) the need to infer latent semantic states that capture the progression of reasoning steps, despite the lack of explicit supervision; (ii) learning stable and generalizable transition dynamics

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s). * Equal Contribution

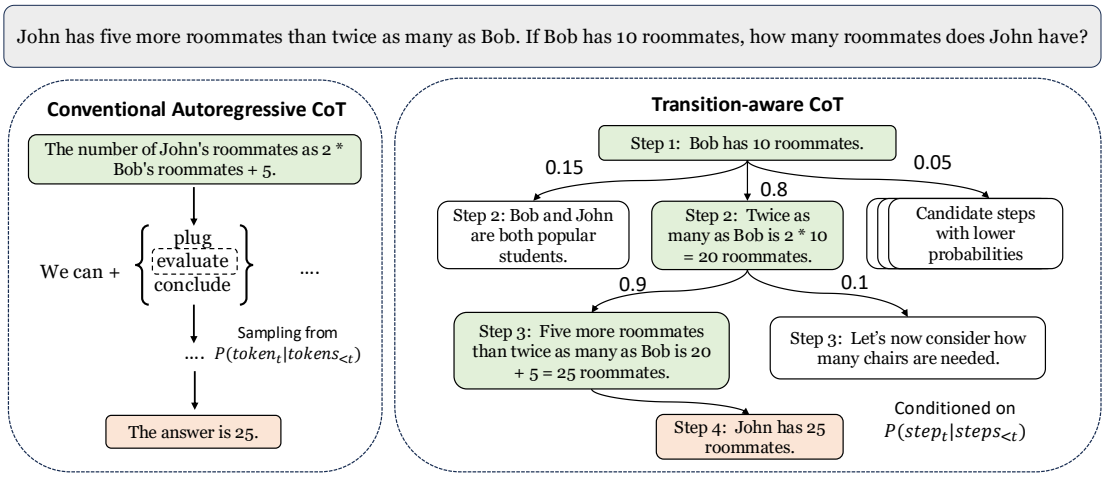



Figure 1: Illustration of the difference between conventional CoT prompting and CTRLS.

across these latent states; and (iii) adapting LLMs to generate coherent reasoning steps conditioned on these latent abstractions. To address these challenges holistically, we propose a unified variational model that jointly learns a latent encoder, a transition policy, and a reasoning adapter for the LLM, optimized under a single **evidence lower bound (ELBO)** objective. This structured design enables semantic grounding and modular reasoning, and naturally supports distributional reinforcement learning (Bellemare et al., 2017; Dabney et al., 2018; Wu et al., 2022) by treating reasoning actions as stochastic policies over latent states. To ensure robust exploration and avoid degenerate trajectories, CTRLS incorporates on-policy optimization with entropy regularization and epsilon-greedy sampling. Overall, our framework provides a principled and tractable foundation for controllable, structured, and adaptive CoT reasoning. We summarize our contributions as follows:

- We introduce a latent-space MDP formulation for explicit modelling of chain-of-thought transitions.
- We propose a distributional reinforcement learning method tailored for robust and explainable exploration in CoT generation.
- We theoretically derive finite-sample evidence lower bounds (ELBO) as the learning objective of CTRLS pre-training.
- We demonstrate empirical improvements in reasoning accuracy, diversity, and exploration efficiency, and showcase enhanced explainability on standard benchmarks.

## 2 RELATED WORK

**Chain-of-Thought Reasoning.** Chain-of-Thought (CoT) prompting improves large language models (LLMs) by encouraging intermediate reasoning steps before producing final answers (Wei et al., 2022; Chu et al., 2023; Wu et al., 2024; Wang et al., 2026). To enhance reasoning quality, prior works introduce self-evaluation (Ling et al., 2023; Shinn et al., 2023) or integrate external knowledge (Zhao et al., 2023). Coconut (Hao et al., 2024) explores reasoning in continuous latent space to bypass token-level constraints. In contrast, we focus on controllability and interpretability via structured latent transitions in language space. Beyond linear CoT, recent work explores structured reasoning, including Tree-of-Thought (ToT) (Yao et al., 2023) and Chain-of-Preference Optimization (CPO) (Zhang et al., 2024), which guide CoT using preferred ToT trajectories. Building on these insights, we model reasoning as transitions over a latent state space, enabling structured step-wise guidance without explicit search.

**Reinforcement Learning for LLM Reasoning.** Reinforcement learning (RL) has been widely used to enhance the reasoning abilities of large language models (LLMs), particularly via Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022; Bai et al., 2022) and Direct Preference Optimization (DPO) (Rafailov et al., 2023; Li et al., 2025a; Wu et al., 2025b; Huang et al., 2025c,b), which learns reward models from human preferences and optimizes LLMs using policy gradient methods such as PPO. While effective for preference alignment, RL-based reasoning faces challenges from sparse and delayed rewards. To address this, recent works introduce outcome-based rewards (Cobbe et al., 2021b; Uesato et al., 2022) or process-based feedback on intermediate steps (Light-

man et al., 2023; Choudhury, 2025), sometimes leveraging verifiers (Su et al., 2025; Mroueh, 2025; Yue et al., 2025; Mundada et al., 2026; **?**) (RLVR) distilled from GPT-4 (Zhang et al., 2024) and off-policy evaluation (Huang et al., 2025a). In contrast, we directly model reasoning dynamics in a latent state space and fine-tune transition behaviours using policy gradients, without relying on external reward models. This enables scalable and interpretable optimization of reasoning trajectories in a self-contained framework.

# 3 PRELIMINARY

## 3.1 Chain-of-thought Reasoning

Chain-of-thought (CoT) reasoning sequentially generates step-by-step intermediate reasoning toward solving complex tasks by large language models (LLMs) (Wei et al., 2022; Kojima et al., 2022). Given an initial prompt or query $x_0$, the LLM policy $\mu$ , iteratively generates reasoning steps $x = (x_1, x_2, \ldots, x_T)$, culminating in a final answer prediction $y$. Each reasoning step $x_t$ consists of sequentially sampled tokens conditioned on all previously generated tokens:

$$x_t \sim \mu(\cdot|x_0, x_{<t}), \quad y \sim \mu(\cdot|x_0, x), \tag{1}$$

where $x_0$ is the input query or prompt (Wei et al., 2022; Wu et al., 2025a; Li et al., 2025b). This autoregressive nature of CoT reasoning presents inherent challenges for controllable generation, as decisions at each step significantly impact subsequent reasoning trajectories. Instead of explicitly modelling this sequential token generation process, we consider reasoning via latent state-transition, where we assume a latent state encoded by an *thought abstraction* model $S_t = \rho_\phi(x_{<t})$ and a state transition process $p_\theta(S_{t+1}|S_t)$.

Latent spaces offer a path to more *explainable* decision making. We take a task-driven view in which intermediate steps are semantically meaningful and form transparent trajectories. Following Yu et al. (2025); Zhang et al. (2025b); Gan et al. (2025); Hou et al. (2023) on steps to human rationale alignment, we further focus on *transition dynamics* among latent reasoning states. Explicitly modeling these transitions makes the process explainable and aligns with our formulation of state-based reasoning and controllable exploration (detailed in Section 3.3).

## 3.2 Distributional Reinforcement Learning

Distributional Reinforcement Learning (DRL) explicitly models uncertainty by representing actions as probability distributions rather than deterministic or discrete selections (Bellemare et al., 2017; Dabney et al., 2018). Specifically, we parametrise the policy $\pi_\theta$ to output Dirichlet distributions over potential next reasoning step in CoTs, which naturally express uncertainty and enable richer exploratory behaviour (Chou et al., 2017). Given state $s_t$, action distributions $a_t$ are drawn as:

$$a_t \sim \pi_\theta(\cdot|s_t), \quad a_t \in \Delta(\mathcal{A}). \tag{2}$$

This formulation inherently captures the uncertainty of stochastic actions over the transition probability distribution of a chain of thoughts. The distributional Bellman operator explicitly incorporates uncertainty into state transitions:

$$\begin{aligned} \mathcal{T}Z(s_t, a_t) &\overset{D}{=} R(s_t, a_t) + \gamma Z(s_{t+1}, a_{t+1}), \\ a_{t+1} &\sim \pi_\theta(\cdot|s_{t+1}) \end{aligned} \tag{3}$$

where $\overset{D}{=}$ denotes distributional equality and $Z(s, a)$ is the return distribution from state-action pairs. Thus, we could enable efficient exploration over CoT transition distribution, essential for improved controllability and explainability in downstream tasks (Haarnoja et al., 2018; Lillicrap et al., 2015).

## 3.3 Formulation: Transition-aware Chain-of-Thought as an MDP

We cast the chain-of-thought (CoT) decoding process into the standard Markov Decision Process (MDP) framework (Bai et al., 2025; Zhang et al., 2025a; Wu et al., 2025a), defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$. This allows us to leverage reinforcement learning algorithms to optimise reasoning trajectories in LLMs.

**State and Transition Dynamics** At each time step $t$, the agent observes a latent reasoning state $s_t \in \mathcal{S}$ that captures the semantic context of the prompt and all previously generated reasoning steps. Formally, we obtain $s_t \sim \rho_\phi(\cdot \mid x_0, x_{<t})$ via a stochastic abstraction model $\rho_\phi : \mathcal{X}_{<t} \to \Delta(\mathcal{S})$, where $\mathcal{X}$ is the space of initial prompts and the sequence of reasoning prefixes (detailed in Section 4.2). Here, $x_0$ is the input query or prompt and $x_{<t} = (x_1, \ldots, x_{t-1})$ are the previously generated reasoning segments. We model latent state evolution with a learned transition matrix $P(s_{t+1} \mid s_t, a_t)$. This kernel encapsulates how one reasoning segment influences subsequent latent abstractions and can be trained jointly with the policy.

**Distributional Action Sampling** Our formulation distinctly leverages a distributional perspective for action sampling within the action space $\mathcal{A}$, which encompasses distributions over admissible reasoning segments rather than discrete or deterministic selections. At each latent reasoning state $s_t$, the policy $\pi_\theta$ explicitly outputs a probability distribution, capturing

the inherent epistemic uncertainty in selecting reasoning segments. This distributional approach facilitates richer exploratory behaviour by enabling the policy to represent a spectrum of plausible reasoning steps, each sampled according to a learned Dirichlet distribution:

$$a_t \sim \pi_\theta(\cdot \mid s_t), \quad a_t \in \Delta(\mathcal{A}), \quad x_t \sim P_\omega(\cdot \mid a_t, x_{<t}),$$

where $\Delta(\mathcal{A})$ denotes the simplex representing the space of action distributions. Here, $\theta$ parametrises a policy network designed to output Dirichlet parameters, thus explicitly modelling uncertainty and providing principled exploration strategies in the CoT action space. $P_\omega$ denotes the adapted LLM with adapted parameters introduced in the backbone LLM $\mu$ for action-conditioned generation, which is explained in detail in Section 4.3.

**Trajectory-level Reward** Aligning with the formulation of LLMs, the quality of the CoT $\tau = (x_1, \ldots, x_T)$ is evaluated by the final answer $y$. Let $y^\star$ denote the ground-truth answer associated with the query $x_0$. The episodic reward is therefore binary $R(\tau, x_0) = \mathbf{1}\{y = y^*\}$. Our learning objective is to maximise the expected terminal accuracy under the controllable policy

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta(\cdot|x_0)}\big[R(\tau, x_0)\big], \tag{4}$$

where $R(\tau)$ is sparse and unbiased, corresponding to the accuracy reported in the evaluation.

# 4 CTRLS: CHAIN-OF-THOUGHT REASONING VIA LATENT STATE TRANSITION

Modeling transition-aware CoT reasoning presents unique challenges: it requires learning latent state representations to capture the semantic progression of reasoning steps, modeling transitions that reflect meaningful and generalizable reasoning dynamics, and conditioning the backbone LLM to generate coherent next steps guided by these latent states. To address these, CTRLS adopts a unified variational framework and an MDP perspective, implementing three core components: (i) a stochastic encoder that abstracts reasoning into latent states via an inference model (Section 4.2), (ii) a state-conditioned UNet that injects latent guidance into token representations and models transitions through a policy network (Section 4.3), and (iii) a two-phase alignment and fine-tuning scheme that combines online pre-training with on-policy reinforcement learning (Section 4.4) (illustrated in Figure 2). This structure supports transition-aware reasoning by optimizing a unified ELBO objective, explicitly modeling uncertainty for principled exploration and controllable, structured generation.

## 4.1 Mathematical Assumptions

**Assumption 5.1 First-order Markov Assumption:** The latent reasoning states follow a first-order Markov process. That is, the transition probability depends only on the previous latent state:

$$P_\theta(z_t \mid x_{<t}, z_{<t}) = P_\theta(s_t \mid s_{t-1})$$

This assumption simplifies the transition dynamics and is justified by the fact that each latent state encodes the full reasoning prefix.

**Assumption 5.2 Autoregressive Factorization of the Variational Posterior:** The variational posterior over latent states is factorized autoregressively to align with the left-to-right generation of LLMs:

$$Q_\phi(z_{1:T} \mid x_{1:T}) = \prod_{t=1}^{T} Q_\phi(z_t \mid x_{\leq t})$$

This ensures consistency between the inference model and the sequential nature of language generation.

## 4.2 Latent State Encoding

Based on our MDP formulation in Section 3.3, we introduce a variational approximation $Q_\phi(z_{1:T}|x_{1:T})$, parametrised by $\phi$, encoding the prompt $x_0$ and reasoning steps $x_{<t}$ into latent state representations.

**Definition 4.1 Inference Model via Variational Posterior** We introduce a variational approximation $Q_\phi(z_{1:T} \mid x_{1:T})$. Consistent with the sequential nature of the data, it factorises autoregressively:

$$Q_\phi(z_{1:T} \mid x_{1:T}) = \prod_{t=1}^{T} Q_\phi\big(z_t \mid x_{\leq t}\big).$$

For each time step $t$, the conditional model $Q_\phi\big(z_t \mid x_{\leq t}\big)$ outputs a probability distribution over the $N$ possible values of $z_t$. It can be instantiated by a Gaussian-mixture posterior, a linear classifier, or a multilayer perceptron (MLP) applied to the hidden representation produced by an encoder.

The autoregressive factorization mirrors the temporal ordering of the observations, enabling the posterior at step $t$ to depend only on the current and past inputs $x_{\leq t}$. By sampling $z_t$ rather than deterministically encoding it, the model captures both semantic content and epistemic uncertainty, reflecting variability in how the system "thinks" before emitting each step.

To instantiate this, we follow (Kveton et al., 2025) to extract token representation $E_t \in \mathbb{R}^{n_t \times d}$ and compute the Gram matrix $G_t = E_t^\top E_t$. Then, to capture the

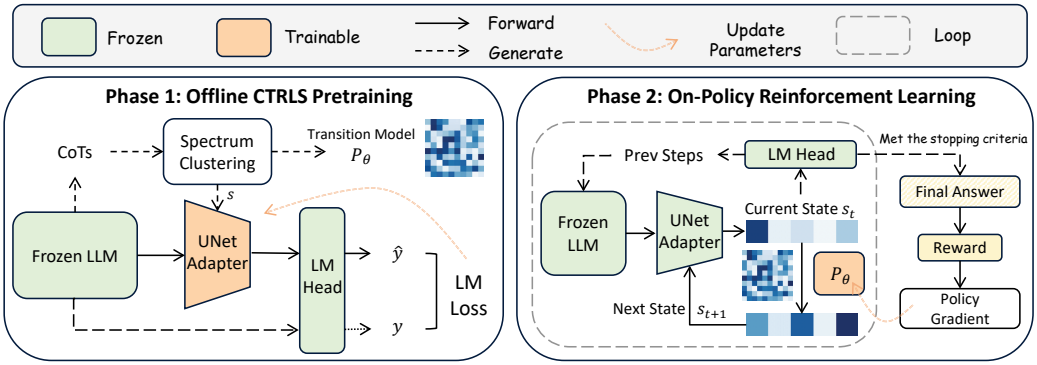



Figure 2: An overview of the proposed two-phase alignment and fine-tuning scheme.

reasoning semantics, we perform spectrum decomposition and take the flattened most informative top $k$ eigenspaces as the representation,

$$e_t := \left[\sqrt{\lambda_1} \cdot \mathbf{q}_1; \ldots; \sqrt{\lambda_k} \cdot \mathbf{q}_k\right] \in \mathbb{R}^{kd}, \tag{5}$$

where $\mathbf{q}_k$ are the corresponding eigenvectors. We apply $k$-means clustering to all such $e_t$ and each reasoning state $s_{t-1} \sim \rho_\phi(\cdot \mid x_{<t})$ is then assigned as the probability distribution corresponding to clustering centroids $\{\gamma_j\}_{j=1}^K$, such that $z_t = \sum_{j=1}^K s_{t,j} \cdot \gamma_j$. This continuous relaxation $Q_\phi$ enables structured reasoning over the latent state space and supports subsequent transition modelling.

### 4.3 State-aware Chain-of-thought Alignment

**Definition 4.2 State-aware Generative Model** Let $\{(x_t, z_t)\}_{t=1}^T$ denote a sequence of observed variables $x_{1:T} \in \mathcal{X}^T$ and latent states $z_{1:T} \in \mathcal{Z}^T$. For parameters $\omega$ (emission) and $\theta$ (transition), the joint distribution factorises autoregressively as

$$P_{\omega,\theta}(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} P_{\omega,\theta}\big(x_t, z_t \mid x_{<t}, z_{<t}\big),$$

where each factor decomposes into a transition term and an emission term

$$\begin{aligned} P_{\omega,\theta}\big(x_t, z_t \mid x_{<t}, z_{<t}\big) = P_\theta\big(z_t \mid x_{<t}, z_{<t}\big) \\ P_\omega\big(x_t \mid x_{<t}, z_{\leq t}\big). \end{aligned} \tag{6}$$

The transition model $P_\theta$ places a prior on the next latent state; it may depend on past observations but is often simplified by the first-order Markov assumption $P_\theta(z_t \mid z_{t-1})$. The emission model $P_\omega$ generates the current observation conditioned on the complete latent trajectory up to time $t$. Together, $P_\theta$ and $P_\omega$ fully specify the stochastic process underlying the data.

**Transition Model** $P_\theta$ To model the conditional latent transition, we first encode the previous reasoning steps into a latent state distribution $s_{t-1} \sim \rho_\phi(\cdot \mid x_{<t})$, as described in Section 4.2. Given that the latent representation $z_t$ is deterministically constructed via $z_t = \sum_{j=1}^K s_{t,j}\gamma_j$ (see Section 4.2), the transition probability naturally reduces from transitions between latent states $z_t$ to transitions between their underlying distributions $s_t$. Specifically, applying the Markov property, we have

$$P_\theta(z_t \mid x_{<t}, z_{<t}) = P_\theta(s_t \mid s_{<t}) = P_\theta(s_t \mid s_{t-1}), \tag{7}$$

where the first equality leverages the deterministic relationship between $z_t$ and $s_t$, and the second equality explicitly invokes the Markov assumption (Wu et al., 2022; Eysenbach et al., 2020).

**Generation Modelling** $P_\omega$ To enable state-aware LLM generation, we design a transformation module that reshapes each token's last hidden state representation $E_{t,i} \in \mathbb{R}^d$ conditioned on a step-wise latent reasoning state $s_t \in \mathbb{R}^{d'}$ based on MDP transition. Specifically, according to the Markov property

$$\begin{aligned} &E'_{t,i} = \mathcal{F}_{\omega_u}\left([\mathcal{F}_{\omega_d}(E_{t,i}); z_t]\right), \quad i = 1, 2, \cdots, n_t \\ &P_\omega(x_t \mid z_{<t}, x_{<t}) = P_\omega(x_t \mid z_{t-1}, x_{<t}) \\ &= \mu(x_t \mid [E'_{1,i<n_1}; \cdots; E'_{t-1,i<n_{t-1}}], \omega), \end{aligned} \tag{8}$$

where $\omega = [\omega_d; \omega_u]$ is a U-Net module that encodes token representation $E_{t,i}$ into a low-rank latent, projects and fuses $z_t$ via a bottleneck interaction, and decodes the result back to dimension $d$. This conditional encoder allows the model to dynamically adjust token representations in sync with the evolving state, with only $O(r(d + d'))$ additional parameters.

Thus, the one-step inference of state-conditional generation in (6) is realised as

$$\begin{aligned} &P_{\omega,\theta}\left(x_t, z_t \mid x_{<t}, z_{<t}\right) \\ &\quad = P_\omega(x_t \mid z_{<t}, x_{<t})\ P_\theta(z_t \mid x_{<t}, z_{<t}) \\ &\quad = \mu(x_t \mid H_{t-1}, \omega)\ P_\theta(s_t \mid s_{t-1}), \end{aligned} \tag{9}$$

$$H_{t-1} := \left[E'_{1,\,i<n_1}; \cdots; E'_{t-1,\,i<n_{t-1}}\right]. \tag{10}$$

where such factorization makes clear how latent dynamics and token sampling interlock. Then we formally introduce the evidence lower bound (ELBO) of the proposed variational model as the objective for model pre-training.

**Theorem 4.3 (Evidence Lower-Bound)** *Consider a latent-state generative model with joint density $P_{\omega,\theta}(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} P_\omega(x_t \mid x_{<t}, z_{\leq t}) P_\theta(z_t \mid x_{<t}, z_{<t})$, and let $Q_\phi(z_{1:T} \mid x_{1:T}) = \prod_{t=1}^{T} Q_\phi(z_t \mid x_{\leq t})$ be any variational distribution. Then, for the learnable parameters $\omega, \theta, \phi$, the marginal log-likelihood of the observations admits the lower bound (detailed derivations in Appendix A)*

$$\mathcal{L}(\omega,\theta,\phi) = \sum_{t=1}^{T} \mathbb{E}_{Q_\phi(z_{\leq t}|x_{\leq t})}[\log P_\omega(x_t \mid x_{<t}, z_{\leq t})] - \sum_{t=1}^{T} \mathbb{E}_{Q_\phi(z_{<t}|x_{\leq t-1})}\Big[\mathrm{KL}\big(Q_\phi(z_t \mid x_{\leq t}) \,\|\, P_\theta(z_t \mid x_{<t}, z_{<t})\big)\Big] \quad (11)$$

*Equality holds if and only if $Q_\phi(z_{1:T} \mid x_{1:T}) = P_\theta(z_{1:T} \mid x_{1:T})$, i.e. when the variational posterior matches the true posterior. Maximising $\mathcal{L}$ therefore constitutes a tractable surrogate objective whose optimisation w.r.t $\omega, \theta, \phi$ simultaneously (i) maximises the data likelihood and (ii) minimises the posterior gap.*

Theorem 5.3 shows that the ELBO provides a tractable way to align latent transitions with reasoning steps. In practice, this means our pretraining objective is not only theoretically sound but also effective: as seen on GSM8K and MATH, optimizing under this objective leads to higher exploration accuracy and fewer spurious steps (detailed later in Section 5.2). Based on the derived ELBO, we propose an offline CTRLS pre-training method in Algorithm 1.

### 4.4 On-Policy Chain-of-thought Reinforcement Learning

After offline pre-training, we further enable on-policy reinforcement learning that optimises only the state-transition model $P_\theta$ through trajectories generated by the current policy. At each decision step $t$, we sample an action distribution $a_t \sim \pi_\theta(\cdot \mid s_t)$ conditioned on the current latent state $s_t$, and subsequently generate the reasoning segment $x_t$ through the state-conditioned LLM generation $P_\omega$. Iteratively repeating this process yields complete trajectories $\tau = (s_t, a_t, s_{t+1})_{t=1}^{T}$ ending with a final predicted answer $y$.

**Trajectory-level Reward and Bellman Function** We evaluate each trajectory $\tau$ based on the correctness of its final answer $y$ against the ground truth $y^*$, providing a binary episodic reward:

$$R(\tau, x_0) = \mathbf{1}\{y = y^*\}. \quad (12)$$

We adopt the distributional Bellman operator (3) to model uncertainty explicitly in state transitions and returns.

**Exploration via Epsilon-Greedy** To effectively balance exploration and exploitation, we implement epsilon-greedy exploration. Specifically, the exploration-enabled action distribution is obtained by mixing the learned Dirichlet distribution with a uniform distribution over actions:

$$\tilde{\pi}_\theta(a|s_t) = (1-\epsilon)\pi_\theta(a|s_t) + \epsilon \mathrm{Uniform}(\mathcal{A}), \quad (13)$$

where $\epsilon \in [0, 1]$ controls the exploration-exploitation trade-off. With probability $\epsilon$, actions are thus sampled uniformly from the action simplex, promoting exploration of diverse reasoning segments, while with probability $1-\epsilon$, actions follow the learned Dirichlet distribution, ensuring exploitation of promising behaviours.

**Entropy Regularization** To further encourage robust exploration and prevent premature convergence to suboptimal solutions, we employ entropy regularization. By adding an entropy-based penalty to the learning objective, the policy maintains diversity in the action distributions, effectively exploring the full action space and discovering potentially more rewarding trajectories. The entropy term is defined as:

$$\mathcal{H}(\pi_\theta(s_t)) = -\sum_{a \in \mathcal{A}} \pi_\theta(a|s_t) \log \pi_\theta(a|s_t). \quad (14)$$

**Overall Learning Objective and Policy Gradient** Integrating trajectory rewards and exploration strategies, our reinforcement learning objective maximises expected trajectory rewards augmented with entropy-based exploration:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta(\cdot|x_0)}\Big[R(\tau, x_0) + \alpha \sum_{t=1}^{T} \mathcal{H}(\pi_\theta(s_t))\Big], \quad (15)$$

where $\alpha$ controls the strength of entropy regularization. Using the REINFORCE estimator, the gradient of this objective with respect to policy parameters $\theta$ is computed as:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\Big[\big(R(\tau) + \alpha \sum_{t=1}^{T} \mathcal{H}(\pi_\theta(s_t))\big) \times \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\Big]. \quad (16)$$

This gradient formulation aligns with accuracy-based evaluation metrics in CoT benchmarks, guiding the optimization of the distributional state-transition model. We further illustrate the on-policy reinforcement learning process in Algorithm 2.

# 5 EXPERIMENTS

## 5.1 Experimental Setup

We conduct experiments on two instruction-tuned language models, LLaMA-3.2-3B-Instruct (Grattafiori et al., 2024) and Qwen2.5-3B-Instruct (Team, 2024). Both models serve as the backbone for integrating our framework without modifying any pretrained weights. We evaluate on two math reasoning benchmarks, GSM8K (Cobbe et al., 2021a) and MATH (Hendrycks et al., 2021), which might require step-by-step CoT generation to solve arithmetic and competition-level mathematics problems, respectively. Implementation details are in Appendix F.

## 5.2 State Transition-aware Exploration

To assess the controllability of our pre-trained, transition-aware chain-of-thought generator, we compare CTRLS against the corresponding base models (LlaMA3.2 and Qwen2.5). In Table 1, we report results for two generation temperatures, $\eta \in \{0.5, 0.7\}$, and an $\epsilon$-greedy exploration strategy with $\epsilon \in \{0.1, 0.3, 0.5\}$. For each test question and each $(\eta, \epsilon)$ configuration, we sample 20 reasoning trajectories from both the base model and CTRLS. We measure **exploration accuracy (pass@20)** as the fraction of questions for which the correct answer appears in at least one of the 20 samples, and **success rate (Succ.)** as the proportion of samples that yield a valid final answer after chain-of-thought generation.

Based on the observations in Table 1, CTRLS consistently outperforms its backbone counterparts in both exploration accuracy and success rate across all datasets. Although performance varies with the choice of $\epsilon$, CTRLS surpasses purely temperature-based sampling in every setting, confirming that explicit state-transition guidance yields more effective exploratory behaviour. Such effective control over latent space exploration is consistent with the theoretical guarantees established in Theorem 4.3. The more informative trajectories produced by CTRLS provide a stronger learning signal for subsequent reinforcement learning, accelerating policy improvement in later training stages.

## 5.3 Impact of Exploration Configurations

In Table 2, we experiment on different exploration configurations and study the effects for on-policy reinforcement learning. **Entropy regularisation.** We observe that raising the entropy weight from $H$=0 to $H$=0.01 consistently broadens the search space. For **LLaMA-3.2**, this translates into a modest but steady gain on both datasets, while **Qwen2.5** shows an even clearer trend, where $H$=0.01 delivers the best accuracy on both datasets, with a minor loss in success rate. These results confirm that a small amount of entropy pressure prevents premature convergence to high-probability but sub-optimal transitions.

**$\epsilon$-greedy sampling.** Injecting stochastic jumps at every step also fosters diversity, but the magnitude of $\epsilon$ matters. For **LLaMA-3.2**, a mild setting ($\epsilon$=0.1) yields the highest overall gains, while a larger perturbation ($\epsilon$=0.3) instead hurts both accuracy and solution validity, echoing the classic explore–exploit trade-off. **Qwen2.5** is more robust for both $\epsilon$ values, yet excessive randomisation still reduces the success rate on the harder MATH dataset.

**Question**: For how many positive integers $n$ is $n^2 - 3n + 2$ a prime number?

**Base (Qwen2.5)**

**Final:** $\boxed{2}$

**Steps:**
1. First, we factorize $n^2 - 3n + 2$. This can be written as $(n - 1)(n - 2)$.
2. For $(n - 1)(n - 2)$ to be a prime number, one of the factors must be 1 and the other must be the prime number itself.
3. Therefore, for $(n - 1)(n - 2)$ to be prime, either $n - 1 = 1$ or $n - 2 = 1$ must hold true.
4. (Wrong conclusion) Thus, there are two values of $n$ that satisfy the condition: $n = 2$ and $n = 3$.

**CTRLS (Qwen2.5)**

**Final:** $\boxed{1}$

**Steps:**
1. (Same factorization) First, we factorize $n^2 - 3n + 2$. This can be written as $(n - 1)(n - 2)$.
2. (Same condition) For $(n-1)(n-2)$ to be a prime number, one of the factors must be equal to 1 because a prime number has exactly two distinct positive divisors: 1 and itself.
3. (Same derivation) Therefore, either $n - 1 = 1$ or $n - 2 = 1$..
4. (Self-reflection) However, we need to check if these values satisfy the condition that $(n-1)(n-2)$ is indeed a prime number.
5. (reduce wrong answer) When $n = 2$, $(n - 1)(n - 2) = (2 - 1)(2 - 2) = 1 \cdot 0 = 0$, which is not a prime number.
6. (Correct conclusion) Thus, there is only one value of $n$ that satisfies the condition, namely $n = 3$.

Figure 3: **Qualitative comparison.** CTRLS correctly verifies candidate solutions and filters out invalid cases, while the baseline fails to check whether the resulting value is truly prime.

## 5.4 Case Study

**Self-reflection** One notable advantage of our transition-based reasoning is its ability to perform self-reflection. As illustrated in Figure 3, both the baseline and CTRLS derive the correct candidate values for $n$ such that $n^2 - 3n + 2$ is prime. However, the baseline prematurely outputs both values as correct without verifying whether the resulting expression is actually a prime number. In contrast, CTRLS continues reasoning by explicitly validating the primality condition

Table 1: Comparison of exploration accuracy and success rate of CTRLS and the base models.

| Generation Temperature | Exploration | LlaMA3.2 GSM8K pass@20 | LlaMA3.2 GSM8K Succ. | LlaMA3.2 MATH pass@20 | LlaMA3.2 MATH Succ. | Qwen2.5 GSM8K pass@20 | Qwen2.5 GSM8K Succ. | Qwen2.5 MATH pass@20 | Qwen2.5 MATH Succ. |
|---|---|---|---|---|---|---|---|---|---|
| $\eta = 0.5$ | Base | 72.5 | 95.0 | 50.0 | 90.0 | 87.5 | **100.0** | **65.0** | 65.0 |
| | $\epsilon = 0.1$ | 77.5 | **100.0** | 47.5 | **95.0** | **90.0** | **100.0** | 57.5 | 60.0 |
| | $\epsilon = 0.3$ | 75.0 | **100.0** | **60.0** | **95.0** | 87.5 | **100.0** | 62.5 | 52.5 |
| | $\epsilon = 0.5$ | **82.5** | **100.0** | 52.5 | 90.0 | 87.5 | **100.0** | **65.0** | **75.0** |
| $\eta = 0.7$ | Base | 77.5 | **100.0** | 47.5 | **97.5** | 80.0 | **100.0** | 72.5 | **67.5** |
| | $\epsilon = 0.1$ | 77.5 | **100.0** | 50.0 | **97.5** | **85.0** | 97.5 | 62.5 | 65.0 |
| | $\epsilon = 0.3$ | 70.0 | **100.0** | **60.0** | **97.5** | 82.5 | 97.5 | 62.5 | **67.5** |
| | $\epsilon = 0.5$ | **85.0** | **100.0** | 47.5 | 90.0 | 80.0 | 97.5 | **77.5** | **67.5** |

Table 2: RL performance under entropy regularization (left block) and $\epsilon$-greedy exploration (right block).

| Model | Entropy | GSM8K pass@20 | GSM8K Succ. | MATH pass@20 | MATH Succ. | $\epsilon$-Greedy | GSM8K pass@20 | GSM8K Succ. | MATH pass@20 | MATH Succ. |
|---|---|---|---|---|---|---|---|---|---|---|
| LlaMA3.2 | Base | 55.6 | 80.0 | 41.0 | 71.8 | Base | 55.6 | 80.0 | 41.0 | 71.8 |
| | $H = 0.0$ | 57.6 | 80.8 | 42.8 | 72.2 | $\epsilon = 0.1$ | 57.4 | 80.4 | 43.0 | 73.4 |
| | $H = 0.01$ | 56.8 | 79.6 | 42.6 | 74.0 | $\epsilon = 0.3$ | 55.8 | 79.4 | 42.0 | 71.2 |
| Qwen2.5 | Base | 64.2 | 91.8 | 22.6 | 33.8 | Base | 64.2 | 91.8 | 22.6 | 33.8 |
| | $H = 0.0$ | 65.8 | 89.6 | 22.8 | 33.4 | $\epsilon = 0.1$ | 66.2 | 89.8 | 22.2 | 32.2 |
| | $H = 0.01$ | 66.4 | 90.2 | 23.6 | 34.6 | $\epsilon = 0.3$ | 66.0 | 90.2 | 23.4 | 34.2 |

Table 3: GPT-4 rubric scores (0–10) on GSM8K with Qwen. Averaged over 100 samples per setting. We evaluate on four different aspects: (**S**)elf-reflection, (**A**)lgebra, (**C**)oherence, (**H**)allucination.

| Setting | S | A | C | H↓ | Overall |
|---|---|---|---|---|---|
| Base | 7.62 | 8.44 | 8.23 | 8.51 | 8.05 |
| $\epsilon = 0.0, H = 0.0$ | 7.82 | 8.83 | 8.43 | 8.66 | 8.34 |
| $\epsilon = 0.1, H = 0.0$ | 7.74 | 8.65 | 8.33 | 8.51 | 8.17 |
| $\epsilon = 0.1, H = 0.01$ | 7.73 | 8.54 | 8.35 | 8.55 | 8.16 |
| $\epsilon = 0.3, H = 0.0$ | 7.80 | 8.77 | 8.45 | 8.68 | 8.30 |
| $\epsilon = 0.3, H = 0.01$ | 7.72 | 8.73 | 8.37 | 8.60 | 8.20 |

for each candidate. It correctly identifies $n = 2$ as invalid, as $(2-1)(2-2) = 1 \cdot 0 = 0$, which is not a prime. This self-reflective validation step leads to the correct final answer. Such state-aware step transitions, which is semantically grounded on explainable states, help avoid early commitment to incorrect conclusions.

**Corrected algebra errors and reduced hallucinated steps** CTRLS enhances symbolic reasoning by reducing common algebraic mistakes—such as misapplied formulas and incorrect substitutions—as well as suppressing spurious steps that lack logical grounding. It preserves variable relationships across steps, avoids unnecessary formalisms, and maintains alignment with the problem's structure. Representative examples are provided in Appendix D.

**LLM-as-a-judge evaluation.** To complement accuracy-based metrics, we additionally include a GPT-4 based rubric evaluation. For each configuration, we randomly sample 100 model outputs on GSM8K and let GPT-4 score them on a 1–10 scale across five criteria: self-reflection, algebra correctness, logical coherence, reduction of hallucinated steps, and overall quality. Table 3 summarises the averaged results for Qwen; detailed prompts and evaluation pipeline are deferred to Appendix E.

## 6 CONCLUSION

We presented CTRLS, a principled framework for transition-aware chain-of-thought reasoning that casts step-wise generation as a latent-state Markov decision process. By modelling reasoning dynamics through distributional policies and optimizing latent transitions via reinforcement learning, CTRLS enables structured, interpretable, and controllable CoT generation. Our experiments demonstrate that CTRLS consistently improves reasoning accuracy, exploration efficiency, and robustness across math benchmarks, and further showcases explainability. Beyond performance, qualitative analyses highlight its ability to recover from symbolic errors, suppress spurious reasoning, and engage in self-reflective correction. We believe CTRLS offers a foundation for more systematic and verifiable reasoning in large language models.

**Acknowledgements**

This work was partially supported by the U.S. National Science Foundation under Grant IIS-2432486.

**References**

Chenjia Bai, Yang Zhang, Shuang Qiu, Qiaosheng Zhang, Kang Xu, and Xuelong Li. Online preference alignment for language models via count-based exploration. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International conference on machine learning*, pages 449–458. PMLR, 2017.

Po-Wei Chou, Daniel Maturana, and Sebastian Scherer. Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution. In *International conference on machine learning*, pages 834–843. PMLR, 2017.

Sanjiban Choudhury. Process reward models for llm agents: Practical framework and directions. *arXiv preprint arXiv:2502.10325*, 2025.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. Navigate through enigmatic labyrinth a survey of chain of thought reasoning: Advances, frontiers and future. *arXiv preprint arXiv:2309.15402*, 2023.

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

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

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, pages 1096–1105. PMLR, 2018.

Benjamin Eysenbach, Swapnil Asawa, Shreyas Chaudhari, Sergey Levine, and Ruslan Salakhutdinov. Off-dynamics reinforcement learning: Training for transfer with domain classifiers. *arXiv preprint arXiv:2006.13916*, 2020.

Zeyu Gan, Yun Liao, and Yong Liu. Rethinking external slow-thinking: From snowball errors to probability of correct reasoning, 2025. URL `https://arxiv.org/abs/2501.15602`.

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

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

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

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

Yifan Hou, Jiaoda Li, Yu Fei, Alessandro Stolfo, Wangchunshu Zhou, Guangtao Zeng, Antoine Bosselut, and Mrinmaya Sachan. Towards a mechanistic interpretation of multi-step reasoning capabilities of language models, 2023. URL `https://arxiv.org/abs/2310.14491`.

Chengkai Huang, Junda Wu, Zhouhang Xie, Yu Xia, Rui Wang, Tong Yu, Subrata Mitra, Julian McAuley, and Lina Yao. Pluralistic off-policy evaluation and alignment. *arXiv preprint arXiv:2509.19333*, 2025a.

Hongtao Huang, Chengkai Huang, Junda Wu, Tong Yu, Julian McAuley, and Lina Yao. Listwise preference diffusion optimization for user behavior trajectories prediction. *arXiv preprint arXiv:2511.00530*, 2025b.

Zihan Huang, Junda Wu, Rohan Surana, Tong Yu, David Arbour, Ritwik Sinha, and Julian McAuley. Image difference captioning via adversarial preference optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33746–33758, 2025c.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language

models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Branislav Kveton, Xintong Li, Julian McAuley, Ryan Rossi, Jingbo Shang, Junda Wu, and Tong Yu. Active learning for direct preference optimization. *arXiv preprint arXiv:2503.01076*, 2025.

Xintong Li, Chuhan Wang, Junda Wu, Rohan Surana, Tong Yu, Julian McAuley, and Jingbo Shang. Importance sampling for multi-negative multimodal direct preference optimization. *arXiv preprint arXiv:2509.25717*, 2025a.

Xintong Li, Junda Wu, Tong Yu, Rui Wang, Yu Wang, Xiang Chen, Jiuxiang Gu, Lina Yao, Julian McAuley, and Jingbo Shang. CoMMIT: Coordinated multimodal instruction tuning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 11522–11536, Suzhou, China, 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.582. URL `https://aclanthology.org/2025.emnlp-main.582/`.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. Deductive verification of chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 36:36407–36433, 2023.

Youssef Mroueh. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025.

Gagan Mundada, Zihan Huang, Rohan Surana, Sheldon Yu, Jennifer Yuntong Zhang, Xintong Li, Tong Yu, Lina Yao, Jingbo Shang, Julian McAuley, et al. Ws-grpo: Weakly-supervised group-relative policy optimization for rollout-efficient reasoning. *arXiv preprint arXiv:2602.17025*, 2026.

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

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

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. Expanding rl with verifiable rewards across diverse domains. *arXiv preprint arXiv:2503.23829*, 2025.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL `https://qwenlm.github.io/blog/qwen2.5/`.

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

Chuhan Wang, Xintong Li, Jennifer Yuntong Zhang, Junda Wu, Chengkai Huang, Lina Yao, Julian McAuley, and Jingbo Shang. Scenealign: Aligning multimodal reasoning to scene graphs in complex visual scenes. *arXiv preprint arXiv:2601.05600*, 2026.

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

Junda Wu, Zhihui Xie, Tong Yu, Handong Zhao, Ruiyi Zhang, and Shuai Li. Dynamics-aware adaptation for reinforcement learning based cross-domain interactive recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 290–300, 2022.

Junda Wu, Tong Yu, Xiang Chen, Haoliang Wang, Ryan Rossi, Sungchul Kim, Anup Rao, and Julian McAuley. Decot: Debiasing chain-of-thought for knowledge-intensive tasks in large language models via causal intervention. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14073–14087, 2024.

Junda Wu, Xintong Li, Ruoyu Wang, Yu Xia, Yuxin Xiong, Jianing Wang, Tong Yu, Xiang Chen,

Branislav Kveton, Lina Yao, et al. Ocean: Offline chain-of-thought evaluation and alignment in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Junda Wu, Rohan Surana, Zhouhang Xie, Yiran Shen, Yu Xia, Tong Yu, Ryan A. Rossi, Prithviraj Ammanabrolu, and Julian McAuley. In-context ranking preference optimization. In *Second Conference on Language Modeling*, 2025b. URL https://openreview.net/forum?id=L2NPhLAKEd.

Yu Xia, Subhojyoti Mukherjee, Zhouhang Xie, Junda Wu, Xintong Li, Ryan Aponte, Hanjia Lyu, Joe Barrow, Hongjie Chen, Franck Dernoncourt, et al. From selection to generation: A survey of llm-based active learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14552–14569, 2025.

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

Sheldon Yu, Yuxin Xiong, Junda Wu, Xintong Li, Tong Yu, Xiang Chen, Ritwik Sinha, Jingbo Shang, and Julian McAuley. Explainable chain-of-thought reasoning: An empirical analysis on state-aware reasoning dynamics. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.904. URL https://aclanthology.org/2025.findings-emnlp.904/.

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

Hongbo Zhang, Han Cui, Guangsheng Bao, Linyi Yang, Jun Wang, and Yue Zhang. Direct value optimization: Improving chain-of-thought reasoning in llms with refined values. *arXiv preprint arXiv:2502.13723*, 2025a.

Shun Zhang, Zhenfang Chen, Yikang Shen, Mingyu Ding, Joshua B Tenenbaum, and Chuang Gan. Planning with large language models for code generation. In *The Eleventh International Conference on Learning Representations*, 2023.

Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *Advances in Neural Information Processing Systems*, 37:333–356, 2024.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, Shuohang Wang, Yelong Shen, and Xin Eric Wang. Soft thinking: Unlocking the reasoning potential of llms in continuous concept space, 2025b. URL https://arxiv.org/abs/2505.15778.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. *arXiv preprint arXiv:2305.03268*, 2023.

Yuchen Zhuang, Xiang Chen, Tong Yu, Saayan Mitra, Victor Bursztyn, Ryan A Rossi, Somdeb Sarkhel, and Chao Zhang. Toolchain*: Efficient action space navigation in large language models with a* search. In *The Twelfth International Conference on Learning Representations*.

## Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

# A Evidence Lower Bound (ELBO) Derivation

We aim to maximize the log-likelihood $\log P_{\omega,\theta}(x_{1:T})$. Since direct maximization is intractable due to the summation over $z_{1:T}$, we maximize the Evidence Lower Bound (ELBO), $\mathcal{L}(\omega, \theta, \phi)$.

Starting from the definition of the log-likelihood and introducing the variational distribution $Q_\phi(z_{1:T}|x_{1:T})$:

$$\begin{aligned}
\log P_{\omega,\theta}(x_{1:T}) &= \log \sum_{z_{1:T}} P_{\omega,\theta}(x_{1:T}, z_{1:T}) \\
&= \log \sum_{z_{1:T}} Q_\phi(z_{1:T}|x_{1:T}) \frac{P_{\omega,\theta}(x_{1:T}, z_{1:T})}{Q_\phi(z_{1:T}|x_{1:T})} \\
&\geq \sum_{z_{1:T}} Q_\phi(z_{1:T}|x_{1:T}) \log \frac{P_{\omega,\theta}(x_{1:T}, z_{1:T})}{Q_\phi(z_{1:T}|x_{1:T})} \quad \text{(Jensen's Inequality)} \\
&= \mathbb{E}_{Q_\phi(z_{1:T}|x_{1:T})} \left[ \log \frac{P_{\omega,\theta}(x_{1:T}, z_{1:T})}{Q_\phi(z_{1:T}|x_{1:T})} \right] \\
&=: \mathcal{L}(\omega, \theta, \phi)
\end{aligned}$$

Now, we expand the ELBO using the factorizations of $P$ and $Q$:

$$\begin{aligned}
\mathcal{L}(\omega, \theta, \phi) &= \mathbb{E}_{Q_\phi(z_{1:T}|x_{1:T})} \left[ \log P_{\omega,\theta}(x_{1:T}, z_{1:T}) - \log Q_\phi(z_{1:T}|x_{1:T}) \right] \\
&= \mathbb{E}_{Q_\phi(z_{1:T}|x_{1:T})} \left[ \sum_{t=1}^{T} \log P_{\omega,\theta}(x_t, z_t|x_{<t}, z_{<t}) - \sum_{t=1}^{T} \log Q_\phi(z_t|x_{\leq t}) \right] \\
&= \mathbb{E}_{Q_\phi(z_{1:T}|x_{1:T})} \left[ \sum_{t=1}^{T} \left( \log P_\omega(x_t|x_{<t}, z_{\leq t}) + \log P_\theta(z_t|x_{<t}, z_{<t}) - \log Q_\phi(z_t|x_{\leq t}) \right) \right] \\
&= \sum_{t=1}^{T} \mathbb{E}_{Q_\phi(z_{1:T}|x_{1:T})} \left[ \log P_\omega(x_t|x_{<t}, z_{\leq t}) + \log P_\theta(z_t|x_{<t}, z_{<t}) - \log Q_\phi(z_t|x_{\leq t}) \right]
\end{aligned}$$

For the term at index $t$, the expectation only needs to be taken over $z_{\leq t}$, as the terms do not depend on $z_{>t}$. Let $Q_\phi(z_{\leq t}|x_{1:T})$ denote the marginal distribution of $z_{\leq t}$ under $Q_\phi(z_{1:T}|x_{1:T})$. With our factorization $Q_\phi(z_{1:T}|x_{1:T}) = \prod_{i=1}^{T} Q_\phi(z_i|x_{\leq i})$, the distribution $Q_\phi(z_{\leq t}|x_{1:T})$ simplifies to $\prod_{i=1}^{t} Q_\phi(z_i|x_{\leq i})$. We use the shorthand $Q_\phi(z_{\leq t}|x_{\leq t})$ as in the outline, representing this product distribution.

$$\begin{aligned}
\mathcal{L}(\omega, \theta, \phi) &= \sum_{t=1}^{T} \mathbb{E}_{Q_\phi(z_{\leq t}|x_{\leq t})} \left[ \log P_\omega(x_t|x_{<t}, z_{\leq t}) + \log P_\theta(z_t|x_{<t}, z_{<t}) - \log Q_\phi(z_t|x_{\leq t}) \right] \\
&= \sum_{t=1}^{T} \left( \mathbb{E}_{Q_\phi(z_{\leq t}|x_{\leq t})} \left[ \log P_\omega(x_t|x_{<t}, z_{\leq t}) \right] + \mathbb{E}_{Q_\phi(z_{\leq t}|x_{\leq t})} \left[ \log P_\theta(z_t|x_{<t}, z_{<t}) - \log Q_\phi(z_t|x_{\leq t}) \right] \right)
\end{aligned}$$

Now consider the second expectation term. We can rewrite the expectation over $z_{\leq t}$ as an expectation over $z_{<t}$ followed by an expectation over $z_t$: $\mathbb{E}_{Q_\phi(z_{\leq t}|x_{\leq t})}[\cdot] = \mathbb{E}_{Q_\phi(z_{<t}|x_{\leq t-1})} \left[ \mathbb{E}_{Q_\phi(z_t|x_{\leq t})}[\cdot] \right]$.

$$\begin{aligned}
&\mathbb{E}_{Q_\phi(z_{\leq t}|x_{\leq t})} \left[ \log P_\theta(z_t|x_{<t}, z_{<t}) - \log Q_\phi(z_t|x_{\leq t}) \right] \\
&= \mathbb{E}_{Q_\phi(z_{<t}|x_{\leq t-1})} \left[ \sum_{z_t} Q_\phi(z_t|x_{\leq t}) \left( \log P_\theta(z_t|x_{<t}, z_{<t}) - \log Q_\phi(z_t|x_{\leq t}) \right) \right] \\
&= \mathbb{E}_{Q_\phi(z_{<t}|x_{\leq t-1})} \left[ -\sum_{z_t} Q_\phi(z_t|x_{\leq t}) \log \frac{Q_\phi(z_t|x_{\leq t})}{P_\theta(z_t|x_{<t}, z_{<t})} \right] \\
&= -\mathbb{E}_{Q_\phi(z_{<t}|x_{\leq t-1})} \left[ \mathrm{KL} \left( Q_\phi(z_t|x_{\leq t}) \parallel P_\theta(z_t|x_{<t}, z_{<t}) \right) \right]
\end{aligned}$$

Substituting this back into the ELBO expression:

$$\mathcal{L}(\omega, \theta, \phi) = \sum_{t=1}^{T} \left( \mathbb{E}_{Q_\phi(z_{\leq t}|x_{\leq t})} \left[ \log P_\omega(x_t|x_{<t}, z_{\leq t}) \right] - \mathbb{E}_{Q_\phi(z_{<t}|x_{\leq t-1})} \left[ \mathrm{KL} \left( Q_\phi(z_t|x_{\leq t}) \parallel P_\theta(z_t|x_{<t}, z_{<t}) \right) \right] \right)$$

This is the final form of the ELBO. It consists of two main terms summed over time:

1. **Expected Reconstruction Log-Likelihood**: The expectation of the log-probability of the observed data $x_t$ given the history and the inferred latent states $z_{\leq t}$.

2. **Expected KL Divergence**: The negative KL divergence between the approximate posterior $Q_\phi(z_t|x_{\leq t})$ and the latent prior/transition model $P_\theta(z_t|x_{<t}, z_{<t})$, averaged over the inferred previous latent states $z_{<t}$. This term acts as a regularizer, encouraging the approximate posterior to stay close to the prior.

Maximizing this ELBO with respect to $\omega$, $\theta$, and $\phi$ trains the model. The expectations are typically approximated using samples from $Q_\phi$.

## B Algorithm of CTRLS Pre-training

Algorithm 1 outlines the pre-training procedure of CTRLS. The model is trained to align step-wise reasoning with latent state transitions using supervised CoT trajectories and associated state embeddings.

**Algorithm 1:** Offline CTRLS Pretraining

**Input:** Dataset $\mathcal{D}$; pretrained LLM $P_\omega$; number of clusters $K$
**Output:** CTRLS generator $P_\omega$; transition model $P_\theta$; inference model $Q_\phi$

```
for each (x, c_{1:T}) ∈ D do
    for each step c_t do
        E_t ← token embeddings from P_ω;
        G_t ← E_t^⊤ E_t ;                                   // Gram matrix
        e_t ← spectrum features of G_t ;                    // Eq. (5)
        Save e_t;
Cluster {e_t} to obtain centroids {γ_1, ..., γ_K};
Compute soft assignments {s_t} for all steps ;              // State distributions
Define Q_φ as the combination of spectral encoder and soft assignment mechanism ;
Optimize P_θ to fit transition pairs (s_{t-1} → s_t) via KL ;   // Eq. (11)
for each (x, c_{1:T}) ∈ D do
    for each step c_t do
        z_t ← Σ_{j=1}^{K} s_{t,j} · γ_j ;
        Inject z_t into token representations ;             // Eq. (8)
    Update P_ω to minimize supervised loss
return P_ω, P_θ, Q_φ
```

## C Algorithm of CTRLS for On-policy Reinforcement Learning

Algorithm 2 details the on-policy reinforcement learning phase. The model fine-tunes its transition policy using trajectory-level rewards and sampled state dynamics to improve reasoning performance.

### C.1 On-policy Reinforcement Learning

Building on the exploration scheme, we fine-tune both backbones on GSM8K and MATH with on-policy RL under four settings: (i) $\epsilon$-greedy only, (ii) entropy regularisation only, (iii) both techniques, and (iv) no exploration (baseline). Figure 4a shows the on-policy learning curves. For LlaMA3.2, entropy regularization is crucial, as the policy collapses to a few high-probability actions and training stalls when it is disabled. CTRLS improves in the early phase (i.e., first 500 steps) without entropy regularisation, but its performance subsequently degrades as action diversity in the transition distribution diminishes.

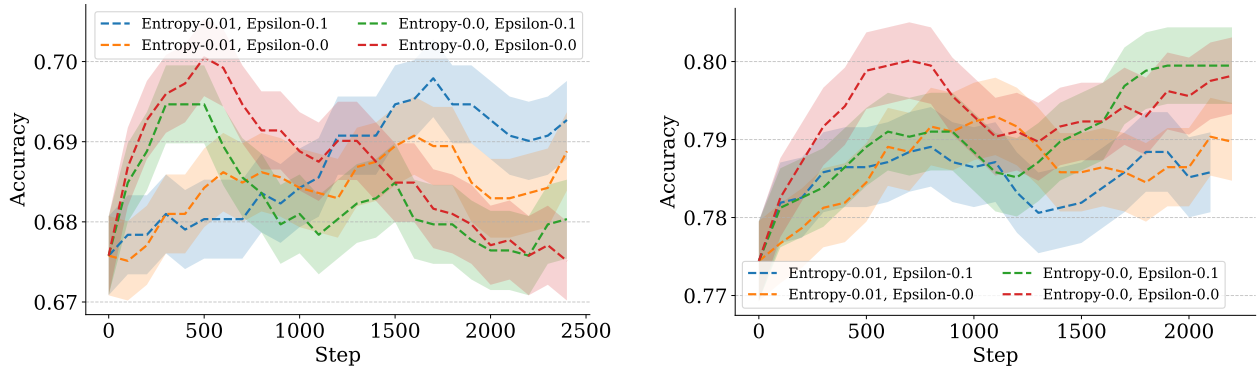
In addition, we observe that entropy regularization prevents this collapse, and combining it with $\epsilon$-greedy sampling yields the fastest and most stable gains, confirming that the two mechanisms are complementary and that

**Algorithm 2:** On-Policy RL Fine-tuning for CTRLS

**Input:** Pretrained generator $P_\omega$; transition model $P_\theta$; inference model $Q_\phi$; centroids $\{\gamma_1, ..., \gamma_K\}$; training dataset $\mathcal{D}$; number of steps $T$

**Output:** Fine-tuned transition model $P_\theta$

**for** *each* $(x, y^*) \in \mathcal{D}$ **do**
  $s_0 \leftarrow Q_\phi(x)$ ; // Initial latent state from prompt
  Set $x_0 \leftarrow x$; trajectory $\tau \leftarrow \emptyset$;
  **for** $t = 1$ *to* $T$ **do**
    $z_t \leftarrow \sum_j s_{t,j} \cdot \gamma_j$ ; // Weighted latent vector
    Inject $z_t$ into token embeddings via Eq. (8) ;
    Sample $x_t \sim P_\omega(\cdot \mid x_{<t}, z_{\leq t})$ ; // State-aware generation
    Append $(s_t, x_t)$ to trajectory $\tau$ ;
    **if** *stopping criterion met (e.g., EOS token)* **then**
      **break**
    Sample $s_{t+1} \sim \epsilon$-Greedy$(P_\theta(\cdot \mid s_t))$ ; // Latent transition (soft dist.)
  Compute answer $\hat{y}$ from full $x_{1:T}$;
  Set reward $R = \mathbf{1}\{\hat{y} = y^*\}$ ;
  Compute policy gradient with entropy bonus ;
  Update $P_\theta$ using optimizer ;
**return** $P_\theta$



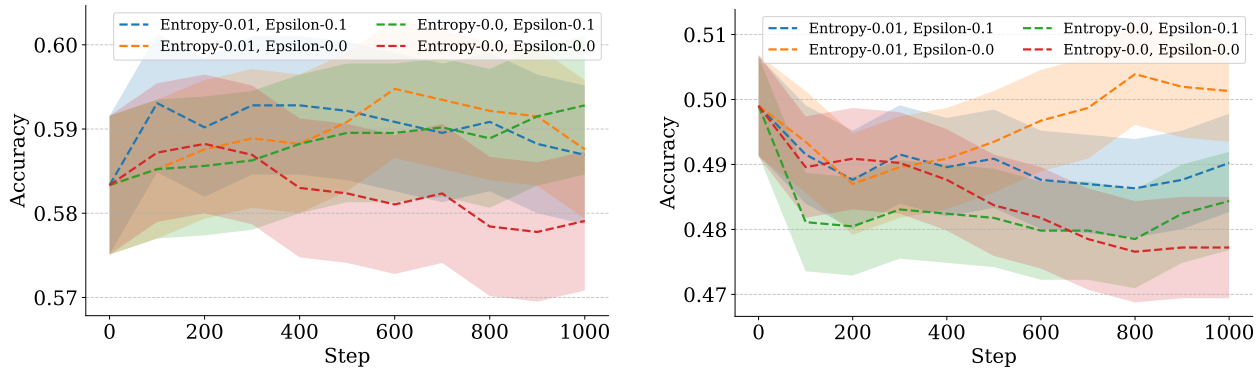

(a) LlaMA3.2

(b) Qwen2.5

Figure 4: On-policy learning curves for CTRLS with **LlaMA3.2** and **Qwen2.5**.

CTRLS benefits most when both are employed. We observe a similar trend for Qwen2.5 backbone model in Figure 4b, while CTRLS achieves more robust on-policy learning improvement even without entropy regularization.

In Figure 5, we further present the on-policy learning of both LlaMA3.2 and Qwen2.5 models of CTRLS on MATH dataset. As discussed in Section C.1, the $\epsilon$-greedy exploration and entropy regularization enhance the learning robustness on LlaMA3.2 based CTRLS. However, we also observe learning degeneration on MATH dataset for Qwen2.5 model. Such problem is derived from the sensitivity of exploration instability discussed in Section 5.3.




(a) MATH

(b) MATH

Figure 5: On-policy reinforcement learning curves for **LlaMA3.2** and **Qwen2.5** on MATH dataset.

## D Case Study

We provide qualitative examples to support the findings discussed in Section 5.4. The Figure 6 illustrates how CTRLS corrects an algebraic substitution error made by the baseline. The Figure 7 shows how CTRLS avoids hallucinated symbolic steps and follows a simpler, task-grounded reasoning path.

## E GPT-as-judge Evaluation Details

We implement an automated evaluation pipeline to quantify qualitative aspects of reasoning. The pipeline is based on the following steps:

1. **Sampling.** For each experiment configuration, we subsample 100 solutions from the model outputs.

2. **Prompt construction.** Each solution is converted into a structured prompt containing: the original question, the ground-truth answer, the step-by-step reasoning, and the model's final answer.

3. **GPT-4 evaluation.** We query GPT-4 (temperature 0.1) with instructions to respond strictly in JSON format. The model assigns scores (1–10) for the following criteria:

   - Self-reflection and validation
   - Correctness of algebra
   - Logical coherence
   - Reduction of hallucinated steps
   - Overall quality

4. **Post-processing.** We parse the JSON output, compute averages, and report aggregate results. When parsing fails (e.g., malformed JSON), we apply simple regex extraction or fall back to default values.

**Question**: Let $a_1, a_2, a_3, \ldots$ be an arithmetic sequence. If $\frac{a_4}{a_2} = 3$, what is $\frac{a_5}{a_3}$?

**Base (LlaMA3.2)**

**Final:** $\boxed{4}$

**Steps:**
1. In an arithmetic sequence, the difference between consecutive terms is constant. Let's denote this common difference as $d$.
2. We can express $a_4$ as $a_2 + 2d$ and $a_5$ as $a_3 + 2d$. Given that $\frac{a_4}{a_2} = 3$, we can write this as $\frac{a_2+2d}{a_2} = 3$.
3. (Wrong derivation) Solving for $d$, we get $2d = 3a_2$, which simplifies to $d = \frac{3}{2}a_2$. Now, we can express $a_5$ as $a_3 + 2d = a_3 + 2\left(\frac{3}{2}a_2\right) = a_3 + 3a_2$.
4. (Wrong conclusion) Therefore, the ratio $\frac{a_5}{a_3}$ is equal to $\frac{a_3+3a_2}{a_3} = 1 + \frac{3a_2}{a_3} = 1 + 3 = 4$.

**CTRLS (LlaMA3.2)**

**Final:** $\boxed{2}$

**Steps:**
1. (Same setup) In an arithmetic sequence, the difference between consecutive terms is constant. Let's denote this common difference as $d$.
2. (Same derivation) We can express $a_4$ as $a_2 + 2d$ and $a_5$ as $a_3 + 2d$. Given that $\frac{a_4}{a_2} = 3$, we can write this as $\frac{a_2+2d}{a_2} = 3$.
3. (Correct derivation) Solving for $d$, we get $2d = 2a_2$, which simplifies to $d = a_2$. Now, we can express $a_5$ as $a_3 + 2a_2$ and $a_3$ as $a_2 + a_2$, which simplifies to $a_3 = 2a_2$.
4. (Correct conclusion) Substituting these expressions into $\frac{a_5}{a_3}$, we get $\frac{a_3+2a_2}{a_2+a_2} = \frac{2a_2+2a_2}{2a_2} = \frac{4a_2}{2a_2} = 2$.

Figure 6: **Qualitative comparison.** CTRLS corrects a symbolic substitution error made by the baseline in an arithmetic reasoning task, resulting in a more accurate final answer.

## F Implementation Details

We follow the prompt design in (Wei et al., 2022) to guide step-wise CoT generation. For each question, we sample multiple CoT trajectories and retain only those yielding the correct final answer. The filtered set forms the training data for CTRLS pretraining. Token embeddings are projected into spectrum embeddings, clustered into $K$=64 latent states, and used to estimate the transition matrix. To encourage diverse reasoning paths, sampling with temperature and top-$k$ filtering is applied during generation. The entire process takes approximately 2 hours on two A6000 GPUs.

**Question**: For how many real values of $x$ is $\sqrt{63-\sqrt{x}}$ an integer?

Base (LlaMA3.2)

**Final:** $\boxed{4}$

**Steps:**

1. We start by letting $n = \sqrt{63-\sqrt{x}}$, which implies that $n^2 = 63 - \sqrt{x}$. This further simplifies to $\sqrt{x} = 63 - n^2$, and squaring both sides gives us $x = (63 - n^2)^2$.
2. (Hallucinated step) We can rewrite this equation as $n^2 + m^2 = 63$, which is a form of Pell's equation.
3. (Misled solution) This gives us the possible values of $n$ as $3, 9, 15$, and $21$, and the corresponding values of $m$ as $6, 0, -6$, and $-12$, respectively.
4. (Wrong conclusion) Therefore, there are $\boxed{4}$ possible values of $x$ for which $\sqrt{63-\sqrt{x}}$ is an integer.

CTRLS (LlaMA3.2)

**Final:** $\boxed{8}$

**Steps:**

1. (Same setup) We start by letting $n = \sqrt{63-\sqrt{x}}$, which implies that $n^2 = 63 - \sqrt{x}$. This further simplifies to $\sqrt{x} = 63 - n^2$, and squaring both sides gives us $x = (63 - n^2)^2$.
2. (Correct step) The perfect squares less than or equal to 63 are $0, 1, 4, 9, 16, 25, 36, 49$.
3. (Correct conclusion) Therefore, the number of real values of $x$ for which $\sqrt{63-\sqrt{x}}$ is an integer is equal to the number of perfect squares less than or equal to 63, which is 8.

Figure 7: **Qualitative comparison.** CTRLS avoids hallucinated symbolic reasoning and correctly counts the number of valid values based on integer square analysis, while the baseline follows an incorrect symbolic path.